Journal of Computational Physics: X 6 (2020) 100051

Contents lists available at ScienceDirect

Journal of Computational Physics: X

www.elsevier.com/locate/jcpx

An implicit local time-stepping method based on cell reordering for multiphase flow in porous media

Gaute Linga [a], Olav Møyner [a], Halvor Møll Nilsen [a], Arthur Moncorgé [b], Knut-Andreas Lie [a,*]

[a] SINTEF Digital, Mathematics and Cybernetics, Oslo, Norway
[b] TOTAL, Reservoir Simulation Department, Pau Cedex, France

ARTICLE INFO

Article history:
Received 26 June 2019
Received in revised form 7 February 2020
Accepted 17 February 2020
Available online 16 March 2020

Keywords:
Numerical flow simulation
Sequential implicit
Flow in porous media
Multiphase flow
Local time-stepping
Adaptivity

ABSTRACT

We discuss how to introduce local time-step refinements in a sequential implicit method for multiphase flow in porous media. Our approach relies heavily on causality-based optimal ordering, which implies that cells can be ordered according to total fluxes after the pressure field has been computed, leaving the transport problem as a sequence of ordinary differential equations, which can be solved cell-by-cell or block-by-block. The method is suitable for arbitrary local time steps and grids, is mass-conservative, and reduces to the standard implicit upwind finite-volume method in the case of equal time steps in adjacent cells. The method is validated by a series of numerical simulations. We discuss various strategies for selecting local time steps and demonstrate the efficiency of the method and several of these strategies by through a series of numerical examples.

1. Introduction

Simulation of multiphase flow in porous media often imparts resolving processes that occur at strongly differing spatial and temporal scales [1,2]. It may therefore be beneficial to adopt inhomogeneous grids, possibly with local adaptive coarsening or refinement, to better resolve details of the flow in specific regions and to capture flow regimes with delicate balances between different physical forces. Increasing the grid resolution or using a higher-order method reduce the local numerical smearing induced by the spatial discretization, whereas *temporal refinement* must be considered to reduce the smearing related to temporal discretization.

Our primary concern herein is simulation of hydrocarbon recovery processes on a reservoir scale, for which variations in pore volumes and intercell flow rates can be several orders of magnitude. In explicit schemes, the time step is restricted by the famous Courant–Friedrichs–Lewy (CFL) condition to ensure stability [3]. This condition is usually very restrictive in reservoir-scale models, and it is therefore more common to use implicit schemes. However, also for these, conventional wisdom from numerical analysis says that the estimated CFL numbers should not be larger than 5–10 to avoid introducing excessive numerical smearing that will result in large pointwise errors in saturations and fluid compositions. One may argue, as we will do later in the paper, that using too large time steps in cells with small pore volumes, so that these fill up too

* Corresponding author.
E-mail addresses: gaute.linga@mn.uio.no (G. Linga), Olav.Moyner@sintef.no (O. Møyner), HalvorMoll.Nilsen@sintef.no (H.M. Nilsen), arthur.moncorge@total.com (A. Moncorgé), Knut-Andreas.Lie@sintef.no (K.-A. Lie).

https://doi.org/10.1016/j.jcpx.2020.100051

quickly, only affects *local* errors and contributes with negligible amounts to the *global* error. Likewise, in many scenarios in enhanced oil recovery, the leading displacement fronts contain strong self-sharpening mechanisms that make these waves less susceptible to numerical smearing than any trailing discontinuities that are linear or weakly nonlinear waves. But even though this provides another argument to increase time-step sizes moderately, the effective time-step sizes in a simulation would still be limited by the cells with the smallest pore volumes. Another problem is high flow in the near-well region. Rapid transients induced when wells are opened or shut to flow can be particularly challenging for (Newton-type) nonlinear solvers, and the standard approach is to reduce the time step for the whole reservoir to ensure convergence of the nonlinear solver, even though the rapid changes in flow properties are localized.

To overcome these challenges, various types of *local time-stepping* strategies have been proposed in the literature, primarily in conjunction with local grid refinement and/or domain decomposition methods. Quandalle and Besset [4] introduced a multiple time-step approach on composite grids (rectangular grid with local mesh refinement) that enabled use of smaller time steps in the refined cells. Ewing et al. [5] analyzed the stability of local time stepping in the context of domain-decomposition methods, developed error estimates, and proved convergence for parabolic equations; see also [6]. Deimbacher and Heinemann [7] developed a windowing technique in which the flow equations are first solved on a global grid with a coarse time step and then subsequently updated using smaller time steps on locally refined grids in the near-well regions; see [8–10] for later extensions and improvements. The *adaptive implicit method* [11–16] enables larger time steps by combining explicit and implicit time stepping: The method uses a standard implicit pressure, explicit saturation (IMPES) scheme in cells with relatively low CFL numbers and a fully implicit scheme in cells with large CFL numbers.

Mc Namara et al. [17] suggested a particularly interesting local time-stepping method that essentially utilizes ideas from the classical Cascade method [18]. The method iterates over three steps: In the first step, the flow equations are solved using a single Newton iteration to obtain a representative pressure field. In the second step, one picks one of the phases and uses the corresponding phase potential values to sort the cells topologically in descending order. In the last step, one chooses a sequence of local time steps independently in each cell. This is done by a trial-and-error algorithm that halves the step size until the nonlinear solver, localized to the cell, converges within a prescribed number of iterations and each computed saturation increment is less than a prescribed tolerance.

Pressure and flow velocities are generally coupled to transported quantities (saturation or chemical compositions), but in many models and flow regimes, it may still be advantageous to reformulate the model equations as a pressure equation and a set of transport equations. This is particularly valid and useful when capillary and gravitational forces are not dominant, and countercurrent flow only occurs in subsets of the domain. The splitting naturally suggests a sequential solution algorithm that solves the pressure equation and the transport equations in separate substeps [19–22]. The two subproblems usually have strongly different mathematical character [23,24]. Pressure equations tend to exhibit elliptic characteristics, in particular for waterflooding problems, whereas transport equations typically have a strong hyperbolic character. Splitting then enables us to utilize specialized methods for the two subproblems. Recently, Carciopolo et al. [25] proposed a local time-stepping method, based on sequentially implicit integration with so-called multirate time stepping [26] for the hyperbolic transport step. (See also [27] for an explicit asynchronous local time-stepping method.)

In this work, we present a new local time-stepping method for sequentially implicit schemes. Our method is similar to [17] in the sense that it relies on topological ordering to infer causality (i.e., unidirectional flow couplings), but whereas [17] rely on (phase) potential ordering of cells [18,28,29], our method assumes that the total velocity field is kept fixed during each pressure step so that we can order cells according to the *total* flux across cell interfaces [30–34]. The nonlinear transport problem then simplifies to solving a sequence of nonlinear ordinary differential equations (ODEs) localized to each cell (or to blocks representing *cycles* of mutually dependent cells), starting at the inflow and consecutively moving downstream. This procedure not only gives local control over the nonlinear solution process, but also scales linearly with the number of degrees of freedom (cells) in the computational domain. It also gives the nonlinear ODE solver freedom to reduce the time step locally in each cell, which often is necessary to accurately resolve rapid transients and achieve convergence for strongly nonlinear flux functions. We present various strategies for dividing the transport step into local substeps and formulate a framework that guarantees that these local time steps are integrated consistently to ensure mass conservation. The result is a sequentially fully implicit method in which the transport equations are integrated cell by cell, or cycle by cycle, in the downstream direction using spatially varying *minor* time steps.

The article is organized as follows. Section 2 presents the underlying multiphase flow model and introduces a one-dimensional version of our method to convey the basic idea. Section 3 presents the general version of our new method for a multidimensional, fully discrete finite-volume scheme. Section 4 reviews and discusses strategies for choosing local time steps (adaptively and by other means). In Section 5, we validate the method on conceptual and simplified examples, but also on a 3D test case that has many features seen in real-life engineering computations. Finally, in Section 6, we conclude and point to future work.

2. Basic concepts

This section introduces the flow equations and describes the basic ideas of our approach to local time stepping in one spatial dimension.

2.1. Basic requirements to a local time-stepping method

The method we seek to formulate is a finite-volume method in which the local time steps will be asynchronously updated in adjacent cells. The main challenge in developing a local time-stepping scheme is to formulate the coupling between states and fluxes defined in local time intervals. To this end, we may pose some fundamental requirements to the method:

- The method should be mass conserving.
- The method should reduce to a standard finite-volume formulation when sampled at uniform time steps.
- The integration schemes should be *invariant* when time steps are rescaled.

The method we present should be applicable to a range of different systems for which the transport can be integrated separately from the flow field, but herein we consider for concreteness a standard Darcy-type model for flow in porous media.

2.2. Multiphase flow model

Standard flow models consist of equations for mass conservation and the multiphase extension of Darcy's law for fluid flow in porous media:

$$\phi \partial_t s_\alpha + \nabla \cdot \mathbf{v}_\alpha = q_\alpha, \tag{1}$$

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K}(\nabla p_\alpha - \rho_\alpha g \nabla z), \tag{2}$$

for phases $\alpha \in \{1, \dots, n\}$. Here, ϕ denotes porosity, s_α is saturation of phase α, $\mathbf{v}_\alpha$ phase velocity, and q_α a source term. The phase mobility λ_α is defined by $\lambda_\alpha = k_{\mathrm{r},\alpha}/\mu_\alpha$, where $k_{\mathrm{r},\alpha}$ is the relative permeability and μ_α the dynamic viscosity of phase α. Moreover, $\mathbf{K}$ is the permeability tensor, p_α phase pressure, ρ_α phase density, g gravitational acceleration, and z the height coordinate. For brevity, we take the flow to be incompressible so that ρ_α is constant. Closure is obtained by requiring that the sum of saturations equals unity, $\sum_{\alpha=1}^n s_\alpha = 1$, and introducing saturation-dependent capillary pressure, $p_i - p_\ell = p_{\mathrm{c},i\ell}(s_1, \dots, s_n)$, where p_c is the capillary pressure function, and $i, \ell \in \{1, \dots, n\}$. For simplicity, we neglect the effect of capillary pressures in the current analysis (and thus skip subscripts on p) and leave this, as well as hysteretic effects and dependence of $k_{r,\alpha}$ on rock type, for further work.

Summing (1) over the phases, we obtain

$$\nabla \cdot \mathbf{v} = q, \qquad \mathbf{v} = -\lambda \mathbf{K} \nabla p + \lambda g \rho \mathbf{k}, \tag{3}$$

where $\mathbf{v} = \sum_{\alpha=1}^n \mathbf{v}_\alpha$ and $q = \sum_{\alpha=1}^n q_\alpha$. Here, we have defined total mobility $\lambda = \sum_{\alpha=1}^n \lambda_\alpha$, the vector $\mathbf{k} = \mathbf{K}\nabla z$, and weighted density $\rho = \sum_{\alpha=1}^n f_\alpha \rho_\alpha$, where $f_\alpha = \lambda_\alpha/\lambda$ is the fractional flow of phase α. This enables us to rewrite (2) as

$$\begin{aligned} \mathbf{v}_\alpha &= -\lambda_\alpha \mathbf{K}\nabla p + \lambda_\alpha \rho_\alpha g \mathbf{k} = f_\alpha \mathbf{v} + \lambda_\alpha g\,(\rho_\alpha - \rho)\,\mathbf{k} \\ &= f_\alpha \mathbf{v} + \lambda g \sum_{\beta=1}^n f_\alpha f_\beta \left(\rho_\alpha - \rho_\beta\right)\mathbf{k} = f_\alpha \mathbf{v} + \delta_\alpha \mathbf{k}, \end{aligned} \tag{4}$$

and consequently (1) transforms to

$$\phi \partial_t s_\alpha + \nabla \cdot (f_\alpha \mathbf{v} + \delta_\alpha \mathbf{k}) = q_\alpha. \tag{5}$$

2.3. Simplified flow problem

For two-phase flow without source terms and gravity effects, the problem is reduced to solving

$$\nabla \cdot \mathbf{v} = 0, \quad \mathbf{v} = -\lambda \mathbf{K}\nabla p \quad \text{(pressure)} \tag{6}$$

$$\phi \partial_t s + \nabla \cdot \big(f(s)\mathbf{v}\big) = 0 \quad \text{(transport)}. \tag{7}$$

Here, we have taken $s_1 = s$ and thus $s_2 = 1 - s$, so that one of the transport equations becomes superfluous, and subsequently skipped the subscript on f_1. The mobility λ in the expression for $\mathbf{v}$ in (6) depends on s. When this dependence is weak, one can reduce computational complexity by solving these two equations sequentially rather than in a coupled manner. Thus, in the forthcoming, we focus on solving (7) with $\mathbf{v}$ taken to be constant in time.

2.4. One-dimensional formulation of the method

For pedagogical purposes, we concentrate first on formulating a one-dimensional (1D) version of the scheme. The example system we consider is composed of cells along an axis x, where, for simplicity and without loss of generality, we consider a total velocity from left to right ($v > 0$). The continuous version of (7) in 1D is

$$\phi \partial_t s + \partial_x \big(f(s) v \big) = 0. \tag{8}$$

For a standard finite-volume method with spatially uniform time steps, the discrete counterpart of (8) is given by

$$\phi_i \frac{s_i^k - s_i^{k-1}}{\Delta t^k} = \frac{F_{i-1/2}^k - F_{i+1/2}^k}{\Delta x_i}. \tag{9}$$

Here, ϕ_i is the porosity in a given cell i of size Δx_i and s_i^k is the cell-averaged saturation in cell i at time step k with length Δt^k. The time step may vary in time, but not in space. Finally, the discrete phase flux $F_{i-1/2}$ approximates $f(s)v$ across the interface between cells i and $i-1$, located at $x_{i-1/2}$. In 1D, and more generally in the absence of gravity and capillary forces, the flow is always cocurrent and hence the fluxes are modelled using standard implicit upwind weighting:

$$F_{i-1/2} = \begin{cases} f(s_{i-1}) v_{i-1/2}, & \text{for} \quad v_{i-1/2} > 0, \\ f(s_i) v_{i-1/2}, & \text{for} \quad v_{i-1/2} \leq 0. \end{cases} \tag{10}$$

Here, we have introduced the average total flux $v_{i-1/2}^k$ across the interface between cells $i-1$ and i during time step k. For incompressible flow, v^k is constant in space and we can assume that $v_{i-1/2}^k = v^k > 0$ for all cells i and time steps k. Hence, information flows from left to right.

To simplify notation, we henceforth consider a given pressure step (or major step) Δt^k, for which the total velocity v^k is computed, as the time interval $\mathcal{I}$ of interest and drop superscript k for brevity. Furthermore, we set $\mathcal{I} = (0, \Delta t]$ without loss of generality. It is then straightforward to introduce a generalization to a finite-volume method with variable time steps in adjacent cells:

$$\phi_i \frac{s_i^j - s_i^{j-1}}{\Delta t_i^j} = \frac{F_{i,\mathrm{L}}^j - F_{i,\mathrm{R}}^j}{\Delta x_i}. \tag{11}$$

Here, superscript j refers to a *substep* (or minor step) that represents a subdivision of the pressure step Δt, local to cell i, such that $\Delta t = \sum_{j=1} \Delta t_i^j$. That is, in cell i we subdivide the time interval $\mathcal{I}$ into M_i sub-intervals $\mathcal{I}_i^j = (t_i^{j-1}, t_i^j]$ with minor indices $j \in \{1, \dots, M_i\}$; these sub-intervals need not be equally long. Thus, we have the observations (dropping subscript i for brevity)

$$\mathcal{I} = \bigcup_{j=1}^{M} \mathcal{I}^j, \quad t^{k,0} = 0 \quad \text{and} \quad t^{k,M^k} = \Delta t. \tag{12}$$

Now, the main difference between the local time-stepping formulation (11) and a single time step in the standard method (9) is the definition of the local fluxes. In particular, since the minor time steps are not synchronized in two adjacent cells, we must differentiate between the fluxes into and out of the face separating them. To this end, the flux *into* cell i across its left edge during minor step j is denoted by $F_{i,\mathrm{L}}^j$, and the flux *out of* cell i on its right edge is denoted by $F_{i,\mathrm{R}}^j$. The latter, outward flux, can trivially be written as:

$$F_{i,\mathrm{R}}^j = f(s_i^j) v, \quad \text{for } t \in \mathcal{I}_i^j. \tag{13}$$

The challenge now is to determine the local inward flux $F_{i,\mathrm{L}}^j$ based on the outward fluxes from the upstream cell, $\{F_{i-1,\mathrm{R}}^m\}$ for $m = 1, \dots, M_{i-1}$. From the fundamental requirement of mass conservation it follows that the inward flux in cell i during the minor time interval $\mathcal{I}_i^j$ should be exactly equal to the *outward* flux from cell $i-1$ *during the same time interval*. In formal terms, this can be expressed as

$$\Delta t_i^j F_{i,\mathrm{L}}^j = \int_{t_i^{j-1}}^{t_i^j} F_{i-1,\mathrm{R}}(\tilde{t}) \, \mathrm{d}\tilde{t}. \tag{14}$$

For a first-order scheme in time, $F_{i,\mathrm{L}}(t)$ and $F_{i,\mathrm{R}}(t)$ are piecewise constant functions of t, such that

$$F_{i,\mathrm{R}}(t) = F_{i,\mathrm{R}}^{j} \quad \text{for } t \in \mathcal{I}_i^j. \tag{15}$$

We define $\mathcal{I}_{i-1/2}^{j,m}$ as the intersection of the two intervals $\mathcal{I}_i^j$ and $\mathcal{I}_{i-1}^m$ (defined at the interface between two adjacent cells i and $i-1$). The size of this overlap interval is expressed as

$$\Delta t_{i-1/2}^{j,m} = \max\left[\min(t_i^j, t_{i-1}^m) - \max(t_i^{j-1}, t_{i-1}^{m-1}),\ 0\right]. \tag{16}$$

Then, we can write (14) as:

$$\Delta t_i^j F_{i,\mathrm{L}}^j = \sum_{m=1}^{M_{i-1}} \Delta t_{i-1/2}^{j,m} F_{i-1,\mathrm{R}}^m. \tag{17}$$

Note that

$$\Delta t_i^j = \sum_{m=1}^{M_{i-1}} \Delta t_{i-1/2}^{j,m}. \tag{18}$$

Upon inserting (18), (17) becomes a weighted mean

$$F_{i,\mathrm{L}}^j = \sum_{m=1}^{M_{i-1}} \psi_{i-1/2}^{j,m} F_{i-1,\mathrm{R}}^m, \qquad \psi_{i-1/2}^{j,m} = \frac{\Delta t_{i-1/2}^{j,m}}{\Delta t_i^j}, \tag{19}$$

where the weights ψ sum to unity for $m = 1$ to M_{i-1}. We observe thus from (19) that the flux into a cell can be regarded as a time-weighted mean of the fluxes out of the preceding cell. Accordingly, (19) can be written as a matrix equation:

$$\mathbf{F}_{i,\mathrm{L}} = \mathbf{\Psi}_{i-1/2} \mathbf{F}_{i-1,\mathrm{R}}, \tag{20}$$

where

$$\mathbf{F}_{i,\mathrm{L}} = [F_{i,\mathrm{L}}^1, \ldots, F_{i,\mathrm{L}}^{M_i}], \quad \mathbf{F}_{i-1,\mathrm{R}} = [F_{i-1,\mathrm{R}}^1, \ldots, F_{i-1,\mathrm{R}}^{M_{i-1}}] \tag{21}$$

are vectors of length M_i and M_{i-1} representing the flux out of and into the cell interface, and the $M_i \times M_{i-1}$ matrix $\mathbf{\Psi}_{i-1/2} = [\psi_{i-1/2}^{j,m}]_{jm}$ contains the temporal weights from (19). The form (20) saves some notation, and it might be useful to also carry calculations on matrix form to exploit vectorization features in interpreted languages such as Python and MATLAB.

2.4.1. Local nonlinear problem

For completeness, we state the local nonlinear problem that should be solved in each cell, ref. (11). The problem is now, in each cell, to solve $R_i^j(s_i^j) = 0$ for s_i^j, where

$$R_i^j(s_i^j) = |\Omega_i| \left[s_i^j - s_i^{j-1}\right] + \Delta t_i^j \left[f(s_i^j) v - F_{i,\mathrm{L}}^j\right]. \tag{22}$$

Here, $|\Omega_i| = \phi_i \Delta x_i$ is the pore volume of cell i, and is like s_i^{j-1}, Δt_i^j, v, and $F_{i,\mathrm{L}}^j$ already given. This nonlinear problem can be solved with, e.g., a stabilized Newton method, or a more robust bracketed solver such as Brent's method, depending on how complicated the flux function $f(s)$ is.

2.4.2. Boundary conditions

Assigning boundary conditions on the left side of the domain is trivial, as the flux into the first cell can be set to $F_{1,\ell}^j = f(s_\Gamma)v$, where s_Γ is the prescribed boundary saturation at time Δt. The outward flux in the rightmost cell requires no further specification, since the fluxes in the transport step are independent of downstream cells in our upwind discretization.

3. General formulation

In this section, we extend our method to general grids in 2D and 3D. As in the previous section, we only present the discretization within a single pressure (or major) time step, which we without loss of generality set as $t \in [0, \Delta t]$. This is done to avoid including an extra superscript k in all temporal quantities.

3.1. Notation

Before we present the formulation of the general method, it is useful to introduce some basic concepts used to represent general unstructured polyhedral grids.

C_i: Cell with index i and pore volume $|\Omega_i|$.

$\mathcal{F}(C_i)$: The set of indices ℓ such that C_ℓ is a neighbor of C_i. Two cells are defined to be neighbors if they share a common interface $\Gamma_{i\ell}$.

$\Gamma_{i\ell}$: Interface between cells C_i and C_ℓ. Note that $\Gamma_{i\ell} = \Gamma_{\ell i}$. The interface *area* $|\Gamma_{i\ell}|$ is only defined for pairs (ℓ, i), for which $\ell \in \mathcal{F}(C_i)$ and correspondingly $i \in \mathcal{F}(C_\ell)$.

$\hat{\mathbf{n}}_{i\to\ell}$: The normal to the interface $\Gamma_{i\ell}$ pointing from cell C_i to C_ℓ. Note that $\hat{\mathbf{n}}_{i\to\ell} = -\hat{\mathbf{n}}_{\ell\to i}$.

$\mathcal{I}_{i,\ell}^{j,m}$: The overlap between two temporal intervals $\mathcal{I}_i^j$ and $\mathcal{I}_\ell^m$, defined between neighboring cells C_i and C_ℓ, i.e., $\mathcal{I}_{i,\ell}^{j,m} = \mathcal{I}_i^j \cap \mathcal{I}_\ell^m$.

$\Delta t_{i,\ell}^{j,m}$: Width of the interval $\mathcal{I}_{i,\ell}^{j,m}$, i.e., the width of the overlap between substep j in cell C_i and substep m in cell C_ℓ:

$$\Delta t_{i,\ell}^{j,m} = \max\Big[\min(t_i^j, t_\ell^m) - \max(t_i^{j-1}, t_\ell^{m-1}),\ 0\Big]. \tag{23}$$

We note the relation $\Delta t_{i,\ell}^{j,m} = \Delta t_{\ell,i}^{m,j}$ and that $\Delta t_i^j = \sum_{m=1}^{M_\ell} \Delta t_{i,\ell}^{j,m}$.

Further, we introduce the following notation to describe the full discretization of multiphase flow of n phases.

$V_{i\to\ell}$: The total flow rate through the interface $\Gamma_{i\ell}$ *from* cell C_i *into* cell C_ℓ. It represents the approximation:

$$V_{i\to\ell} \approx \int_{\Gamma_{i\ell}} \mathbf{v}(\mathbf{x}, 0) \cdot \hat{\mathbf{n}}_{i\to\ell}\, \mathrm{d}A. \tag{24}$$

In practice, it is given by solving the pressure equation (3) on a discrete grid. Note that $V_{i\to\ell} = -V_{\ell\to i}$. Furthermore, for incompressible flow, we must have

$$\sum_{\ell\in\mathcal{F}(C_i)} V_{i\to\ell} = q_i \approx \int_{C_i} q(\mathbf{x}, 0)\, \mathrm{d}V, \tag{25}$$

where q_i is the integrated source term in cell i. There are multiple ways of obtaining these fluxes, the most usual is by a two-point flux approximation (see e.g., [24]).

$q_{\alpha,i}$: The source term of phase α in cell i is defined analogously to q_i and is constant during the transport step.

$s_{\alpha,i}^j$: Cell-averaged saturation of phase α in cell C_i at *minor* time step j. For a first-order scheme, it approximates s at the centroid of C_i.

$F_{\alpha,i,\ell}^j$: Flux of phase α *into* cell C_i *from* cell C_ℓ (i.e., across $\Gamma_{i\ell}$) during *minor* step j.

3.2. Algorithm

In a general multidimensional framework, the algorithm is a straightforward generalization of that in one dimension. The essential difficulty lies in coupling the fluxes across all interfaces. Another complicating factor is the potential presence of regions of countercurrent flow or other types of *cycles*, i.e., mutually dependent cells that must be solved for simultaneously.

To describe the general algorithm, we assume that the pressure solution step has been carried out, so that all total flow rates $V_{i\to\ell}$ are given and taken to be constant during the transport step. Then the algorithm reads:

1. Order cells according to the total fluxes $V_{i\to\ell}$, such that

$$V_{i\to\ell} > 0 \quad \Leftrightarrow \quad i < \ell \quad \text{for all} \quad \ell \in \mathcal{F}(C_i), \tag{26}$$

for all cells C_i. The effect of this ordering is to permute the discrete nonlinear transport equations to lower triangular form so that they can be solved cell-by-cell in the same way as we discussed in the 1D case in Section 2.4. Note that the requirement (26) will have to be relaxed when dealing with cycles of mutually dependent cells, the treatment of which we defer until later.

2. For all $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M_i\}$, solve sequentially, with respect to s_i^j:

$$s_{\alpha,i}^j - \frac{\Delta t_i^j}{|\Omega_i|} \sum_{\ell\in\mathcal{F}(C_i)} F_{\alpha,i,\ell}^j = s_{\alpha,i}^{j-1} + q_{\alpha,i}\Delta t_i^j, \tag{27}$$

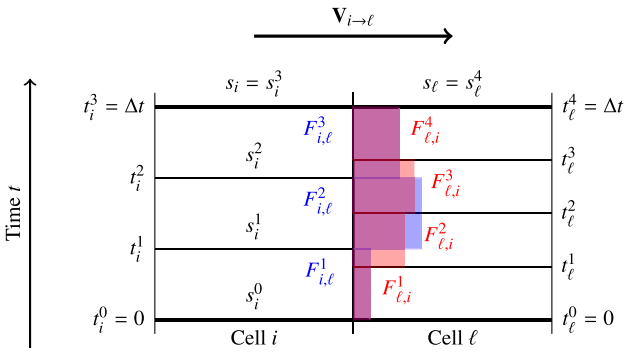

Fig. 1. Piecewise constant reconstruction of local-in-time half-face fluxes that ensures mass conservation. The three blue rectangles represent flux *out* of Cell i, whereas the four red rectangles represent flux *into* Cell ℓ.

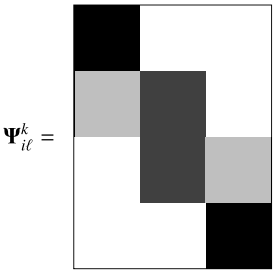

Fig. 2. Visualization of the overlap matrix corresponding to Fig. 1, which gives the flux vector through $\mathbf{F}_{\alpha,i,\ell} = \mathbf{\Psi}_{i,\ell}\mathbf{F}_{\alpha,\ell,i}$ (definition in the text) when $V_{i\to\ell}^k > 0$. Darker shade of an entry signifies higher value.

for $\alpha \in \{1, \dots, n\}$, where the upwind-weighted fluxes can be written as

$$F_{\alpha,i,\ell}^j = \begin{cases} -\frac{1}{\Delta t_i} \sum_{m=1}^{M_\ell} \Delta t_{i,\ell}^{j,m} F_{\alpha,\ell,i}^m, & \text{for} \quad V_{i\to\ell} < 0, \\ -f_\alpha(s_{\alpha,i}^j) V_{i\to\ell}, & \text{for} \quad V_{i\to\ell} \geq 0. \end{cases} \tag{28}$$

The construction of these half-face fluxes is schematically shown in Fig. 1. We can also write this on matrix form,

$$\mathbf{F}_{\alpha,i,\ell} = \begin{cases} -\mathbf{\Psi}_{i,\ell}\mathbf{F}_{\alpha,\ell,i}, & \text{for} \quad V_{i\to\ell} < 0, \\ -\mathbf{f}_{\alpha,i} V_{i\to\ell}, & \text{for} \quad V_{i\to\ell} \geq 0, \end{cases} \tag{29}$$

where

$$\mathbf{F}_{\alpha,i,\ell} = [F_{\alpha,i,\ell}^1, \dots, F_{\alpha,i,\ell}^{M_i}]^T, \qquad \mathbf{\Psi}_{i,\ell} = \frac{1}{\Delta t_i}\mathbf{T}_{i,\ell}, \qquad \mathbf{T}_{i,\ell} = [\Delta t_{i,\ell}^{j,m}]_{jm},$$

$$\mathbf{f}_{\alpha,i} = [f_\alpha(\{s_{\alpha,i}^1\}_{\alpha=1}^n), \dots, f(\{s_{\alpha,i}^{M_i}\}_{\alpha=1}^n)].$$

Here, $\mathbf{\Psi}_{i,\ell}$ is an $M_i \times M_\ell$ matrix. For the schematic problem shown in Fig. 1, the overlap matrix $\mathbf{\Psi}_{i,\ell}$ is shown in Fig. 2. Note that $\mathbf{T}_{i,\ell} = (\mathbf{T}_{\ell,i})^T$, so the overlap matrix need only be computed once per interface $\Gamma_{i\ell}$.

Compared to a standard finite-volume scheme, the essential new ingredient in our method is the need to compute the overlap time steps $\Delta t_{i,\ell}^j$, or equivalently, the overlap matrices $\mathbf{T}_{i,\ell}$. If the time step resolution does not change throughout a simulation, i.e., M_i is the same for all major steps, these overlaps (or overlap matrices) may be precomputed and reused at every major time step. However, local time-stepping schemes can be particularly powerful when adapted to changes in total velocity field and transported quantities. In this case, the local time steps cannot be precomputed, but must be computed on the fly, as discussed in the next section.

Gravity, cycles, and cross-flow. In general, the flux graph represented by $V_{i\to\ell}$ will contain circular dependencies due to cross-flow, e.g., as a result of countercurrent flow caused by gravity, mutual dependencies because of capillary effects, or because the pressure equation is discretized by a multipoint (and non-monotone) scheme. Herein, we use the word *cycle* to refer to such sets of mutually dependent cells. One approach for solving cycles would be to iterate around each cycle until convergence. A more practical approach is to consider a cycle as a *chunk* of cells to be solved simultaneously. The time steps

within this chunk must then be synchronized and will be solved self-consistently using a standard finite-volume scheme, except that the upstream fluxes will be computed according to the varying time discretization in different cells.

The main challenge with including gravity effects is thus not the appearance of cycles, but that gravity may cause the flow ordering (i.e., the signs of the local phase fluxes) to change and differ between the phases during the transport integration. One way to overcome this potential problem is to split the transport step in two so that the first substep accounts for transport according to Darcy fluxes and the second accounts for gravity segregation; this is a standard approach in streamline simulation [35] and has also been used to extend reordered methods to flow with strong countercurrent effects [32]. One could also imagine using a nonlinear Gauss–Seidel approach in which the transport step is reiterated if flow ordering has changed, but more research is needed to determine whether this is viable or not.

4. Local time-step selection

Pressure steps are usually determined from the imposed report steps and a suitable *a priori* subdivision of these. A second approach is to attempt a nominally small pressure step, followed by a standard chopping procedure that selects subsequent pressure steps based on observed or estimated changes in saturation and composition. One may also choose the pressure step to ensure that the number of nonlinear pressure iterations meets a prescribed target. Local time steps for the transport equation, on the other hand, are typically chosen to limit the amount of numerical smearing or to resolve sharp fronts in particular regions of the simulation domain. This section outlines various strategies for choosing such local time steps.

4.1. Static region-based selection

A simple and straightforward strategy for local adaption is to impose refined time steps in certain sub-regions of the domain. One example would be to use more steps in near-well regions or in high-flow region between two wells by explicitly marking all cells that should have a higher number of minor steps than the rest of the domain. Likewise, for grids with spatially varying resolution caused by adaptive refinement or coarsening, the local time step could be set to match the local grid resolution, e.g., if a local 2×2 refinement is applied to a patch of cells, an accompanying refinement could be imposed on the time steps in the same cells.

4.2. Selection based on local CFL number

A variant of the local approach is to base the minor time step on an estimate of the local CFL number η_i in cell C_i. Following [13], we can estimate this, *based on the flux function*, by

$$\eta_i^j = \Lambda_i^j \bar{\eta}_i. \tag{30}$$

Here, $\bar{\eta}_i$ is the local CFL number for passive, linear advection,

$$\bar{\eta}_i^j = \frac{\Delta t_i^j}{|\Omega_i|} \sum_{\ell \in \mathcal{F}(C_i)} [(V_{i\to\ell})_+ - (q_i)_-], \tag{31}$$

and Λ_i^j is the largest eigenvalue of the Jacobian

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial s_1} & \cdots & \frac{\partial f_{n-1}}{\partial s_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial s_{n-1}} & \cdots & \frac{\partial f_{n-1}}{\partial s_{n-1}} \end{bmatrix} \tag{32}$$

evaluated at $s_{\alpha,i}^j$ for all $\alpha \in \{1, \ldots, n\}$. For brevity, we write $(\cdot)_+ = \max(\cdot, 0)$ and $(\cdot)_- = \min(\cdot, 0)$. Notice also that the sum in (31) only includes flux *out of* the cell. The linear CFL number (31) represents a unit reference frame for local wave propagation speeds and can be computed independently of the actual wave structure for a given flux field.

The user should then specify a CFL target η_{spec}, and use (30) to solve $\eta_i^j(\Delta t_i^j) = \eta_{\text{spec}}$ with respect to Δt_i^j to define the minor time step dynamically in each cell. A constraint on explicit schemes would typically be $\eta_{\text{spec}} \leq 1$. For 1D linear advection ($f_\alpha = s_\alpha$) with constant porosity, this coincides with the CFL number that equalizes contributions from spatial and temporal discretization to the numerical smearing. For nonlinear displacements, the convexity/concavity of the flux function introduces self-sharpening effects that tend to counteract numerical smearing, and experience shows that the estimated CFL number can be up to one order of magnitude above unity before numerical smearing starts to significantly affect the resolution of fronts.

Selecting time steps based on local CFL numbers has the advantage that temporal resolution is automatically increased in high-flow regions, e.g., near wells, and this reduces the need for gradually ramping up the global time step, which is a well-known trick of the trade. A problem with (30), however, is that the Jacobian (32) is computed explicitly so that the

maximum wave speeds used to estimate the time step lag behind the actual wave speeds experienced during the time step. When solving the transport equations cell by cell, we can mitigate this by using information from updated states in the upstream cells to gain a better estimate of the maximum wave speed

$$\tilde{\Lambda}_i = \max_{\substack{\ell \in \mathcal{F}(C_i) \\ V_{i\to\ell}<0}} \left\{ \max_{m=1}^{M_\ell} \left| \frac{f_\alpha(s^m_{\alpha,\ell}) - f_\alpha(s^{m-1}_{\alpha,\ell})}{s^m_{\alpha,\ell} - s^{m-1}_{\alpha,\ell}} \right| \right\}, \tag{33}$$

and use this to define an alternative CFL estimate, $\tilde{\eta}_i^j = \tilde{\Lambda}_i \overline{\eta}_i^j$.

4.3. Selection based on representative CFL numbers

To select regional time steps that balance errors caused by the temporal discretization and errors inherent to the spatial discretization, we propose to transform the cell-wise CFL numbers for passive linear advection (31) into time-of-flight coordinates, so that (31) represents the ratio between the time step and the average incremental travel time $\Delta\tau_i$ over all flow paths that cross the cell,

$$\overline{\eta}_i^j = \frac{\Delta t_i^j}{\Delta\tau_i}, \qquad \Delta\tau_i = \frac{|\Omega_i|}{\sum_{\ell\in\mathcal{F}(C_i)}(V_{\ell\to i})_-}. \tag{34}$$

Here, the average incremental travel time, also referred to as the residence time of the cell, is defined as the ratio between the cells pore volume and the flux *into* the cell.

To see the relative size of cell C_i in Lagrangian coordinates, we must scale the incremental travel time by the average global residence time τ_i of all flow paths that pass through the cell, which can be computed rapidly for each major step by solving a (block) triangular system; see [30,24]. For pure cocurrent flow, the ratio $\Delta\tau_i/\tau_i$ represents the fraction of the total travel time from inlet to outlet that is spent traversing cell C_i.

Our hypothesis is now that we can study the distribution of Lagrangian cell sizes $\{\Delta_i\tau/\tau_i\}$ for the whole reservoir or for a given subregion, pick the mean of this distribution, and insert the corresponding incremental travel time into (34) to compute a *representative CFL number*. This is based on a relatively simple physical argument. For a standard first-order, implicit scheme, the amount of numerical smearing we add to a propagating discontinuity increases proportionally to the local CFL number, and the main effect of using a large time step in a cell with small pore volume or high throughput is that we flood this cell too early (provided that the nonlinear solver is able to converge the time step). However, if premature flooding only takes place in cells that are quickly traversed by the displacement front, the overall adverse effect will usually be small. In other words, a too large local time step can contribute to a large local error measured in s, but the increase of the ϕs-error is small, and the error in s tends to die out as soon as the displacement front enters cells with larger incremental travel times. Thus, a regional time step should be chosen based on a *representative CFL target* and not on the worst-case local CFL number arising in cells with small pore volumes or high throughput. This argument is admittedly quite hand-wavy, but a variety of test cases show that it holds true as long as the representative CFL target is chosen to be unity or slightly above unity.

All over, this gives us an unbiased method for determining the local (in time) CFL number based on the *instantaneous flow field*, and provides a more *physically sound criterion* for choosing time steps than, e.g., using criteria based solely on numerical considerations, such as the convergence of nonlinear solvers.

4.4. Predictor-corrector approach

Using the residence time $\Delta\tau_i$ defined in the previous section, we can rewrite a user-specified CFL restriction per cell as,

$$\Delta t_i^j \le \eta_{\text{spec}} \Delta\tau_i / \Lambda_i^j. \tag{35}$$

Using this nonlinear inequality (remember that Λ_i^j depends upon the unknown solution), we can formulate an iteration scheme in which we accept the local time-step size Δt_i^j if (35) is satisfied. If not, we reduce the time step and solve the local cell (or chunk) again, recheck that the reduced time step and the computed solution satisfy (35), and if necessary, repeat this procedure until the inequality is fulfilled, as outlined in more detail in the next subsection. This approach is very robust, but requires us to compute maximum/minimum eigenvalues of the numerical flux function, which can be expensive.

4.5. Selection based on local saturation updates

A more direct (and somewhat ad hoc) approach, which does not require computation of eigenvalues, is to control the change in saturation locally. In particular, we prescribe an upper limit on the saturation change $\Delta s_{\max}$ allowed during a (minor) time step

$$\Delta s_i^j \leq \Delta s_{\max}, \tag{36}$$

where $\Delta s_i^j = \|s_{\alpha,i}^j - s_{\alpha,i}^{j-1}\|_p$ for a suitable ℓ^p norm. Following the argument from Section 4.3, it may in many cases be more natural to measure the change in volume fraction, $\Delta(\phi_i s_i^j)$. Likewise, for multicomponent flows, one should track changes in component concentrations or component fractions. For brevity, we only discuss Δs in this subsection, since the other methods are essentially the same, except for the obvious modifications.

A straightforward predictor-corrector approach can be used to achieve this (see [36, pp. 54] and [37]). For each cell C_i, we use an iteration of the following form:

1. Let $r = 0$. (If we are in the very first major step, let $\Delta t_i^{0,0} = \Delta t$.)
2. Solve the local problem (27) with the estimated time step $\Delta t_i^{j,r}$ to obtain $s_{\alpha,i}^{j,r}$, which represents a cell-wise saturation change $\Delta s_{\alpha,i}^{j,r} = \|s_{\alpha,i}^{j,r}(\Delta t_i^{j,r}) - s_{\alpha,i}^{j-1}\|_p$.
3. Compute a suggestion for a new time step:

$$\Delta t_i^{j,*} = \frac{(1+\omega)\Delta s_{\max}}{\Delta s_i^{j,r} + \omega \Delta s_{\max}} \Delta t_i^{j,r}, \tag{37}$$

   where $\omega \in [0, 1]$ is a tunable parameter [36].
4. If $\Delta s_i^{j,r-1} < \Delta s_{\max}$, *decline* the local time step. That is, set $\Delta t_i^{j,r+1}$ to the minimum of $\Delta t_i^{j,*}$ and the time remaining part of the major time step, increase r by 1, and go to Step 2. Otherwise, *accept* the local time step, i.e., assign

$$\begin{aligned} \Delta t_i^j &= \Delta t_i^{j,r}, \\ s_{\alpha,i}^j &= s_{\alpha,i}^{j,r}, \quad \text{for } \alpha \in \{1, \ldots, n\}. \end{aligned} \tag{38}$$

   If Δt_i^j is long enough to bring us to the end of the major time step number k, set $\Delta t_i^{0,0} = \min(\Delta t_i^{j,*}, \Delta t^k)$ and go to next cell. Otherwise, set $\Delta t_i^{j+1,0}$ to the minimum of $\Delta t_i^{j,r}$ and the remaining part of the major step, increase j by 1, and go to Step 1.

Note that the most important part of this iteration procedure is the regulator (37), which is commonly used to select global time steps in reservoir simulation. The rest is mainly a procedure to ensure that the local time steps synchronize with the pressure updates.

4.6. Combined approach for cache efficiency

Computational efficiency on a modern computer is usually memory bound, i.e., limited by how fast we can feed the computational cores with sufficient data. Using a cell-by-cell solution procedure, in which we can specify the time step and control the nonlinear iteration process locally in each cell, is computationally optimal in the sense that it minimizes the number of floating-point operators, but will in practice represent an efficiency barrier. To utilize cache more efficiently and decrease latency, we therefore process *chunks* of cells at a time; preferably these cells should also be represented as contiguously in memory as possible to minimize access costs. Doing so inevitably gives larger local systems, more local time steps, and more nonlinear iterations, but reduces the average start-up cost per cell, utilizes cache better, and leaves computational resources less idle. Hence, adapting the time step to n cells simultaneously will typically be significantly faster than adapting the time step individually in n cells.

5. Numerical simulations

In this section, we investigate the effectiveness on a series of representative numerical test cases. We have provided a minimal Python implementation in 1D as Supplementary Material to demonstrate the basic principles. However, the general method for multidimensional polyhedral grids is implemented in the MATLAB Reservoir Simulation Toolbox (MRST) [24], which is an open-source code for MATLAB/GNU Octave, aimed at rapid development of numerical methods for reservoir simulation. Our implementation is built on top of a reordered, cell-by-cell solver presented by Klemetsdal et al. [34], which relies on processing of chunks for efficiency, and uses automatic differentiation to linearize and solve the nonlinear equations. Although our framework is general and is applicable to a wide class of multiphase and multicomponent models under the assumption of cocurrent flow, we herein only present test cases with incompressible two-phase flow in the absence of gravity and capillary effects.

5.1. Validation: 1D Buckley–Leverett displacement

The Buckley–Leverett displacement is a standard benchmark problem for two-phase Darcy flow. In 1D and in the absence of gravity and capillary effects, the formulation (8) holds with constant $v \geq 0$. This problem is exactly solvable, and thus

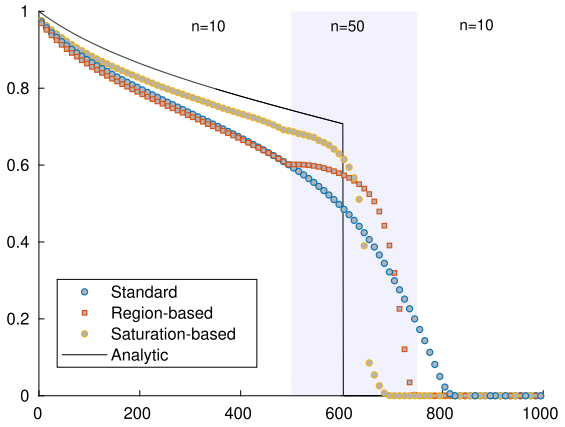

Fig. 3. Buckley–Leverett displacement. Comparison of the exact solution for $M = 1$, a standard implicit upwind scheme with 10 time steps, and the local time-stepping scheme with two different time-step selections. The region-based criterion refines the time step by a factor 5 in the region $x \in [500, 750]$ (light blue), whereas the predictor-corrector scheme with saturation-based criterion imposes a maximum saturation change $\Delta s_{\max} = 0.1$ everywhere.

provides a direct way to assess the accuracy of the method. We consider an imbibition scenario, with $s(0, t) = 1$ (pure water) injected on the left hand side and initial condition $s(x, 0) = 0$ (pure oil). Further, we consider a fractional flow function with quadratic relative permeabilities,

$$f(s) = \frac{s^2}{s^2 + M(1-s)^2}, \tag{39}$$

where $M = \mu_1/\mu_2$ is the viscosity ratio between phase 1 (water) and phase 2 (oil). The exact solution is given by [24]

$$s(x,t) = \begin{cases} g^{-1}(x/t), & \text{for} \quad x/t \le g(0), \\ 0, & \text{for} \quad x/t \ge g(0), \end{cases} \tag{40}$$

where g is the derivative of the convex envelope of f on the interval $[0, 1]$

$$g(s) = \begin{cases} f'(s^*), & \text{for} \quad 0 \le s \le s^*, \\ f'(s), & \text{for} \quad s^* \le s \le 1, \end{cases} \tag{41}$$

and $s^* = \sqrt{M/(1+M)}$ is the saturation at the leading edge of the shock.

Fig. 3 compares the exact solution with approximate solutions computed using a standard upwind finite-volume scheme and our new scheme with region-based or saturation-based refinement of time steps on a uniform grid with 100 cells. The region-based scheme uses 10 time steps in all cells, except for $x \in [500, 750]$, where the time step is refined by a factor 5. Altogether, this amounts to a total of $75 \cdot 10 + 25 \cdot 50 = 2000$ local time steps and gives a similar profile as the standard method in the unrefined part of the domain, but a sharper front in the region of temporal refinement, which leads to more correct prediction of time to water breakthrough. The distinct kink at the intersection between the unrefined and the refined parts of the domain is consistent with the jump in the numerical diffusion coefficient for the modified equation of the scheme. In particular, the numerical smearing goes from being dominated by Δt in the unrefined region, to being dominated by Δx in the refined region. Qualitative artifacts like this are obviously undesirable in practice, and we emphasize that this particular example is a contrived setup that was carefully designed to exaggerate the effect for illustration purposes.

Our second approach relies on the predictor-corrector scheme described in Section 4.5 to limit the local saturation updates to less than $\Delta s_{\max} = 0.1$, and uses a total of 2057 local time steps. This gives a much better approximation to the exact solution using approximately the same total number of local time steps as the region-based approach. Fig. 4 shows a space-time diagram of the displacement profile up to dimensionless time 0.8 PVI, as computed on a grid with 64 cells. Here, we clearly see how the minor steps cluster around the evolving displacement front so that most of the computational effort is spent on adaptive refinement in the parts of the domain where the solution changes most from one time step to the next.

Fig. 5 illustrates that the user-prescribed value of Δs should be related to the strength of the wave one seeks to resolve, meaning that Δs generally should be chosen larger for a favorable displacement with $M > 1$ than for an unfavorable displacement with $M < 1$. In our experience, choosing $\Delta s = \delta \cdot s^*$ for $\delta \in [0.05, 0.15]$ seems to strike a reasonable compromise between accuracy and computational cost. For larger values of Δs, the computed solution is not only much more smeared, but will in many cases contain kinks or buckles that are qualitatively incorrect. With smaller values of Δs, one ends up introducing many local steps without improving the resolution of the displacement front significantly. For multicomponent

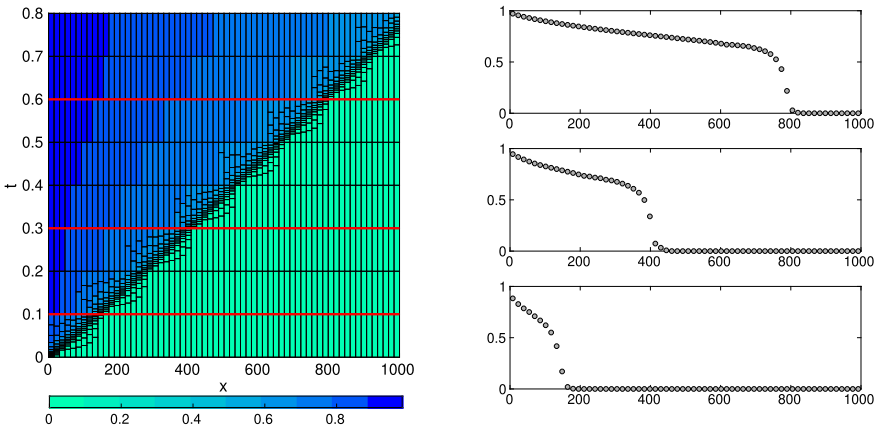

Fig. 4. Space-time diagram of the Buckley–Leverett displacement problem computed using saturation-based selection of local time steps with $\Delta s_{\max} = 0.1$ on a grid with 64 uniform cells. The (x, t) diagram to the left shows how the eight major steps are subdivided adaptively along the displacement front, with colors representing the saturation in each cell at the end of each minor/major time step. The three saturation profiles in the right column are sampled at times indicated by the red horizontal lines in the left-hand plot.

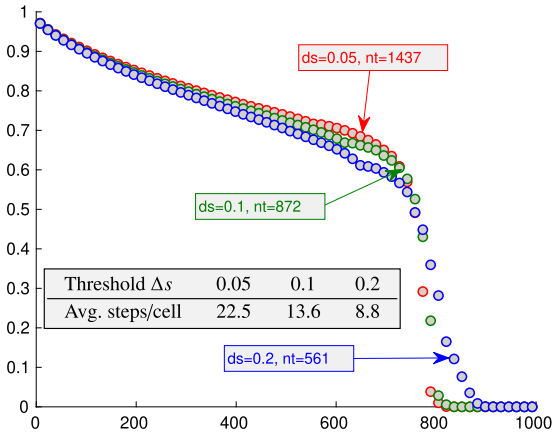

Fig. 5. Buckley–Leverett displacement profile computed with the saturation-based selection of local time steps with three different tolerances, $\Delta s = 0.05$, 0.1, 0.2. The basic setup has 64 uniform cells and 6 major time steps, giving a total of $6 \cdot 64 = 384$ local steps.

or compositional flows, it is important to use a refinement criterion that detects local changes induced by all the waves one wants to resolve accurately.

5.2. Quarter-five spot

A second standard test is the quarter-five spot case, in which we consider a two-dimensional domain $\Omega = [0, L] \times [0, L]$ with constant injection at $(0, 0)$ and production at (L, L) and no-flow conditions across the whole boundary. Fig. 6 shows a direct comparison between the solutions at 40% injected pore volume computed with the saturation-based scheme ($\Delta s_{\max} = 0.1$) and by a standard scheme with no temporal refinement. We clearly see that the local time-stepping scheme captures the sharp injection front more accurately. Fig. 6c reports the local number of substeps M_i required by the adaptive scheme, which clearly shows that most of the effort is spent in the region around the front, as in the 1D Buckley–Leverett case. Fig. 6d reports the ratio between declined (guessed) and accepted substeps. Interestingly, this ratio is highest at the trailing edge of the refinement region, which coincides with the cells having the steepest saturation gradient at the end of the previous major step. This indicates that additional work should be invested in ensuring a better prediction of acceptable steps.

Excessive numerical smearing can exaggerate fingering effects in heterogeneous displacements, as illustrated in Fig. 7. Here, we have sampled petrophysical properties from the top layer of Model 2 from the 10th SPE Comparative Solution Project (SPE 10) [38]. Introducing local time steps clearly reduces the smearing and delays the predicted water breakthrough.

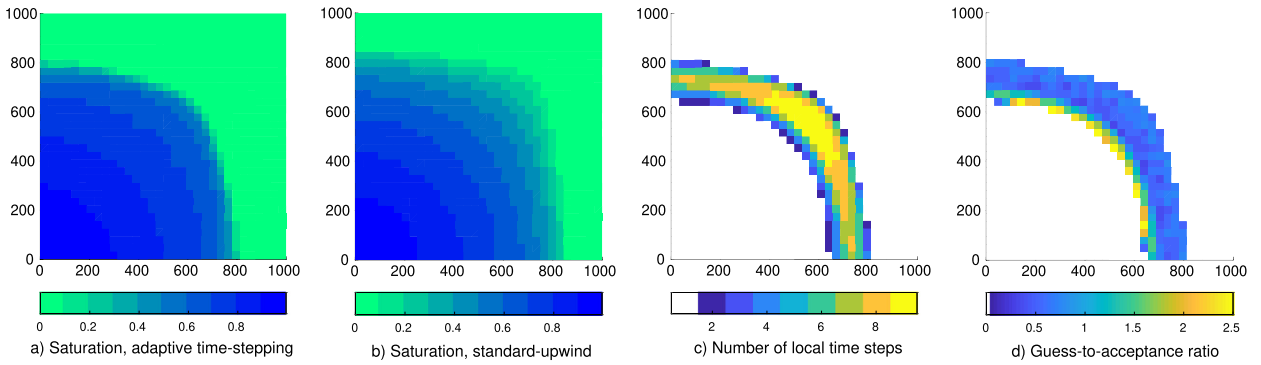

Fig. 6. Homogeneous quarter-five spot test case. (a) Saturation field at time 0.4 PVI computed with the adaptive, saturation-based method. (b) Saturation field computed with the standard upwind method. (c) The number of minor time steps used during the major time step leading up to the field shown in (a). (d) Ratio between declined and accepted time steps associated with (a) and (c).

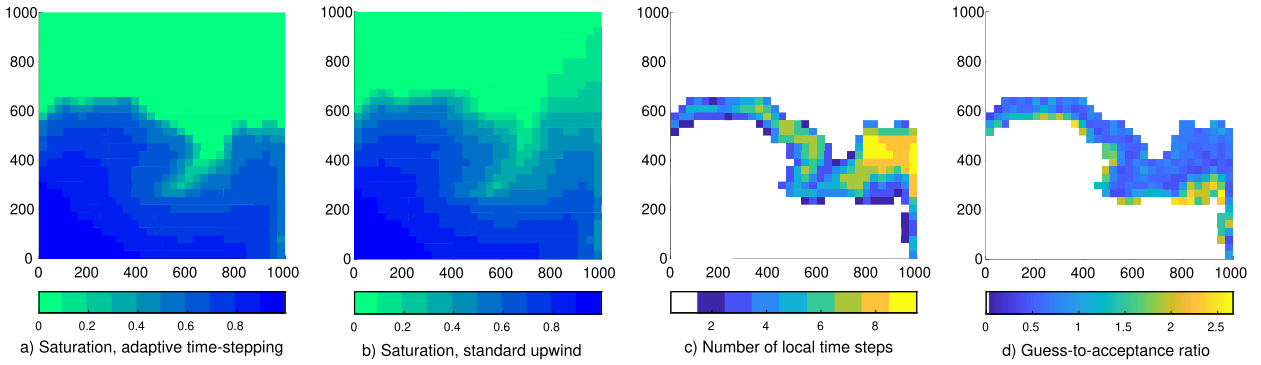

Fig. 7. Heterogeneous quarter-five spot test case. (a) Saturation field at time 0.5 PVI computed with the adaptive, saturation-based method. (b) Saturation field computed with the standard upwind method. (c) The number of minor time steps used during the major time step leading up to the field shown in (a). (d) Ratio between declined and accepted time steps associated with (a) and (c).

5.3. Asymmetry test

To better illustrate the effect of local time steps, we consider a test case with a symmetric flow pattern between one injector and two producers. That is, in a rectangular domain $\Omega = [0, 1000] \times [0, 500]$, we place an injector at $(500, 0)$, one producer at $(0, 500)$ and one at $(1000, 500)$. We select time steps based on the local CFL number, but introduce an asymmetry by setting a CFL target of 1 in the left half of the domain ($x < 500$) and a CFL target of 10 in the other half ($x > 500$). The top panel of Fig. 8 shows a snapshot from the simulation after 50% of the pore volume has been injected. Using more time steps retards the tip of the displacing water and gives an evident lack of symmetry, but with a unit mobility ratio and no gravity effects, the asymmetry does not cause significant movement of water from one half domain to the other.

The associated production curves presented in the bottom panel of Fig. 8 confirm that using more time steps delays the water breakthrough but also predicts a significantly higher overall water production in the left half of the domain. Fig. 9 reports the number of local time steps used in each cell during the first pressure step. This spatial distribution of local time steps only exhibits minor changes throughout the next nine pressure steps.

In the east half-domain, time steps are hardly reduced at all since most cells stay below the CFL target of 10. Hence, the predicted well response for the northeast producer is largely similar to that of both producers in the reference case with ten major steps and no local refinement. The time step is only refined in two semi-circular regions near the injector and producer, respectively, where the flow is almost radial and significantly faster than in the rest of the domain (where most of the residence time of any flow path is accumulated). The plot of cell-wise CFL numbers shows that the effective CFL numbers vary from slightly less than ten along the main diagonal to less than one in the stagnant regions in the northwest and southeast corners.

The situation in the west half-domain is similar: in the high-flow zone along the diagonal, and away from the near-well regions, the pressure step is subdivided into five minor steps to meet a CFL target of 1.0. The required number of extra minor steps gradually decays to zero as we move away from the main diagonal and toward the stagnant zones in the southwest and northeast corners of the half-domain. Likewise, the number of minor steps increases rapidly up to 20 as we approach the wells.

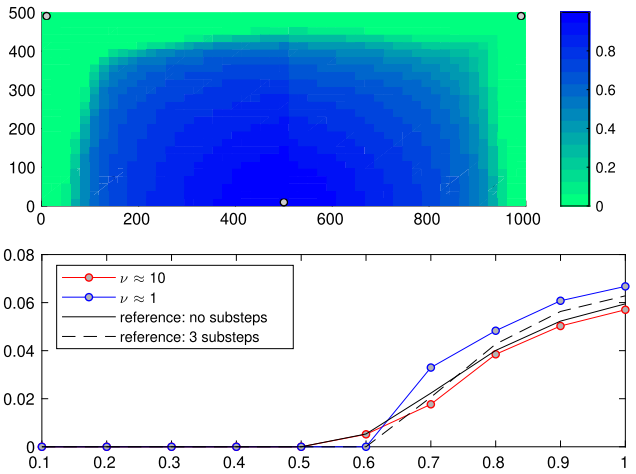

Fig. 8. Asymmetry test with a homogeneous domain with a symmetric well pattern (injector at the midpoint on the south perimeter and producers in the northeast and northwest corners), but with different CFL targets in the east and west half domains. Top: saturation field after 50% injected pore volume. Bottom: Water production curves. The reference solutions are from a standard sequential-implicit solver with ten major steps and either no temporal refinement of the transport steps or a uniform subdivision into 3 minor steps in all cells.

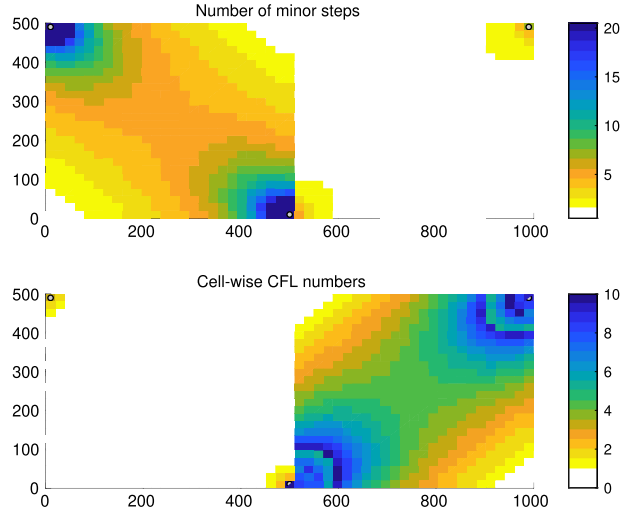

Fig. 9. Number of local time steps and effective CFL numbers in each cell during the first pressure step for the asymmetry test case.

Let us also see how this translates to the *representative CFL number* introduced in Section 4.3. Fig. 10 shows the distribution of Lagrangian cell sizes $\Delta\tau/\tau$, spanning values from $8 \cdot 10^{-4}$ to 0.14 with a mean value of 0.026. Mapping this into the distribution of *linear* CFL numbers gives a value of 5.0, which for a flux function with quadratic relative permeabilities and unit viscosity ratio corresponds to a *nonlinear* CFL number of 10 (since $\Lambda = 2$). The second reference solution reported in Fig. 8 uses three minor steps uniformly in all cells, which would correspond to setting targets of 5/3 and 10/3 on the *linear* and *nonlinear* CFL numbers. With this subdivision, we predict the same time to water breakthrough, but a somewhat lower water production. Interestingly, increasing the uniform subdivision to 5 or 10 steps has very small effect on the production curve. This behavior is representative of other test cases we have run. We interpret this to indicate that whereas the concept of representative CFL numbers can be used to identify reasonable compromises between computational cost and numerical smearing in a standard method, using *locally adapted* time steps has a much more pronounced effect on the overall accuracy of the simulation.

5.4. Conceptual 3D model with nonuniform grid coarsening

In the last example, we consider a conceptual 3D model of an oil-filled anticline that overlies a slightly bent aquifer. Petrophysical data are sampled from the shallow-marine Tarbert formation from the SPE 10 benchmark. Oil is produced from two wells placed at the top of the anticline, each operated at a fixed bottom-hole pressure of 100 bar. Additional pressure support is provided by four wells placed around the peripheral of the anticline structure, each operating at a

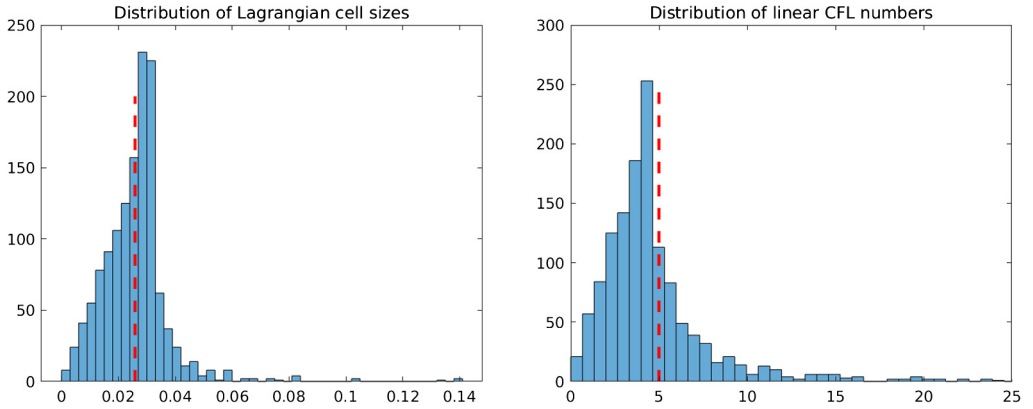

Fig. 10. Distributions used to determine a representative CFL number for the asymmetric test case. The distribution of linear CFL numbers has a long tail, with minimum value 0.22 and maximum value 132.6.

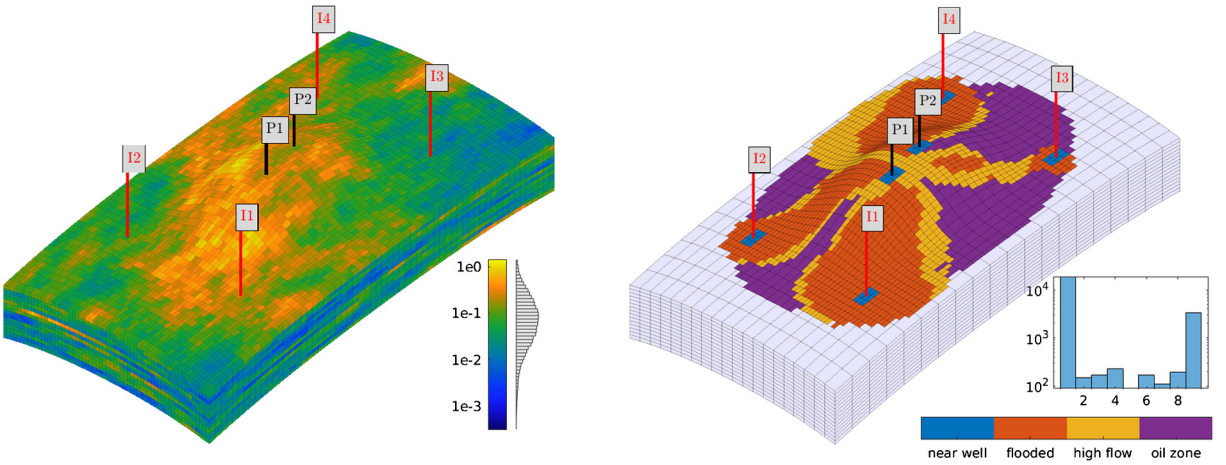

Fig. 11. Conceptual model of an anticline. The left plot shows the reservoir geometry and the well pattern, with cells colored by the lateral permeability on a logarithmic scale. The right shows the coarsened grid, with colors indicating the classification used to determine the local cell resolution. The histogram reports $N(i)$, where $N(i)$ is the number of coarse blocks that contain i fine cells.

fixed bottom-hole pressure of 125 bar. To accelerate simulation of the $60 \times 60 \times 15$ corner-point model, we introduce a nonuniform coarsening [39] that reduces the number of cells by approximately 60% (see the right plot in Fig. 11):

- In all cell neighbors of the perforated well cells, we keep the original fine resolution.
- We also retain the fine resolution in all cells that are likely to be flooded by the water injectors or that lie in a high-flow region of the reservoir. To determine these cells, we first compute time-of-flight τ from injectors and total residence times for all cells. Assuming a two-phase fluid model with quadratic Corey exponents and equal viscosities, the flooded cells are set to be all cells in which τ is less than 1.5 times the total time horizon T for the simulation, $\tau < 1.5T$. Likewise, the high-flow region is set to be those cells in which the total residence time between injector and producer is less than twice the median residence time of all cells perforated by wells.
- All other cells above the initial oil-water contact are coarsened by a factor $2 \times 2 \times 1$, whereas the remaining aquifer cells are coarsened by a factor $3 \times 3 \times 1$.

The right plot in Fig. 12 reports production curves predicted by five different simulations:

ref: The fine grid with 60 uniform pressure steps of length 18.25 days, but with a 4-level geometric ramp-up during the first 18.25 days.

base: The fine grid with 15 uniform pressure steps of length 73 days, except for the first 110 days, which follow the same ramp-up as **ref**.

LTS: Same setup as **base**, except that we use the predictor-corrector method with $\Delta s = 0.1$ for temporally-adaptive, local time stepping.

cbase: The coarse grid with the same time stepping as for **base**.

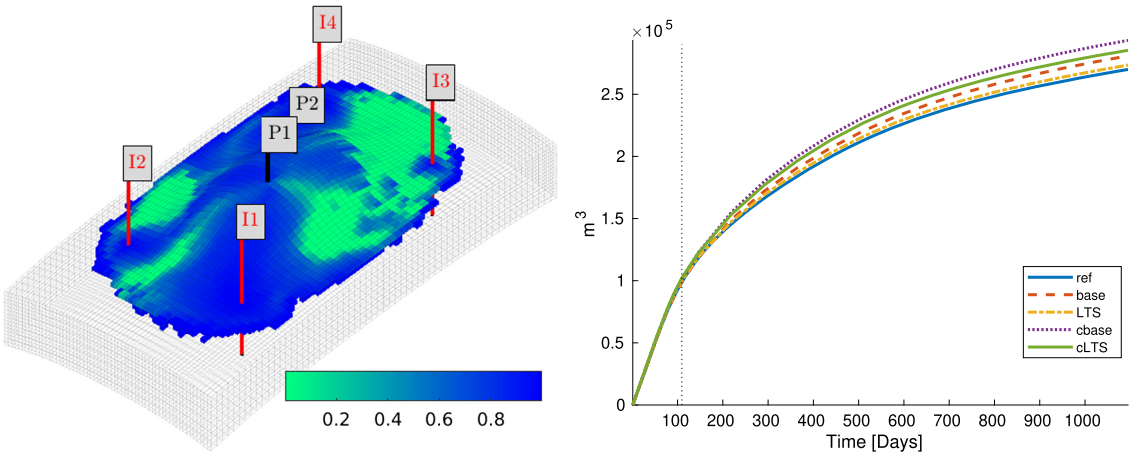

Fig. 12. The left plot shows water saturation in all cells that contain some oil at the end of simulation as computed with the **ref** simulation. The right plot reports cumulative production of oil from producer P1 for all the five simulations.

cLTS: Same setup as **cbase**, except that we statically refine the time step by a factor 5 above the oil–water contact and by a factor 10 in the near-well, flooded, and high-flow coarse blocks.

For efficiency, the local time steps are introduced on chunks of 100 cells at a time. The **base** simulation overestimates the cumulative oil production compared with **ref**. Introducing temporally-adaptive, local time stepping in the transport solve reduces the discrepancy significantly: for producer P2, the cumulative oil production is virtually identical for **ref** and **LTS**. This indicates that the numerical smearing introduced in the transport step is the main error source for the **base** simulation. Coarsening the grid leads to further overestimation in the **cbase** simulation, but introducing static, local time stepping brings the discrepancy for producer P1 closer to that of the **base** simulation on the fine grid. For producer P2, the cumulative oil production from **cLTS** and **base** are virtually identical.

6. Concluding remarks

We have introduced a new approach to adaptive, local time stepping that combines sequential splitting of pressure and transport, flow-based ordering to ensure that the transport equations can be solved cell-by-cell for cocurrent flow, and temporal refinement of the transport step localized to individual cells. We have described a variety of strategies that can be used to select local time steps. Among these, we emphasize the concept of a representative CFL number, based on the mean cell volume in Lagrangian coordinates, which seems to give a good compromise between computational efficiency and numerical smearing. The main advantage of this representative CFL number is that it also can be applied to standard transport solvers without local time-stepping capabilities. However, one can observe significantly better efficiency in terms of accuracy versus computational costs by adapting the time step locally in each cell, but this currently assumes that the flow is fully cocurrent, so that cells (and cycles of mutually dependent cells) can be solved one by one, or grouped into chunks to improve cache efficiency. Local adaption is particularly important to accurately resolve fast flow in near-well regions.

All test cases presented herein are restricted to two-phase flow for simplicity, but our local time-stepping framework is straightforward to extend to other flows for which reordering methods can be utilized to infer causality (unidirectional flow). We believe that our method also can be extended to cases with countercurrent flow induced by gravity, but more research remains to verify this. The method is by design applicable to general unstructured grids, which particularly includes composite grids and most types of local grid refinements. In particular, the framework is readily applicable in the adaptive, multilevel coarsening method introduced recently by Klemetsdal et al. [40,41], which also includes the use of higher-order discontinuous Galerkin spatial discretizations that contribute to further reduce numerical smearing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge Total E&P Norway for funding this research and for granting us permission to publish the results. We also thank August Johansson and Øystein Klemetsdal for stimulating discussions on the development of the method.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcpx.2020.100051.

References

[1] J. Bear, Dynamics of Fluids in Porous Media, Courier Corporation, 2013.
[2] M.J. Blunt, Multiphase Flow in Permeable Media: A Pore-Scale Perspective, Cambridge University Press, 2017.
[3] K.H. Coats, Impes stability: selection of stable timesteps, SPE J. 8 (2003) 181–187, https://doi.org/10.2118/84924-PA.
[4] P. Quandalle, P. Besset, The use of flexible gridding for improved reservoir modeling, in: SPE Reservoir Simulation Symposium, 15-18 November, San Francisco, California, Society of Petroleum Engineers, 1983.
[5] R.E. Ewing, R.D. Lazarov, P.S. Vassilevski, Finite difference schemes on grids with local refinement in time and space for parabolic problems I. Derivation, stability, and error analysis, Computing 45 (1990) 193–215, https://doi.org/10.1007/BF02250633.
[6] B.A. Boyett, M.S. El-Mandouh, R.E. Ewing, Local Grid Refinement for Reservoir Simulation, Computational Methods in Geosciences, vol. 33, Society for Industrial and Applied Mathematics, 1992, pp. 15–28.
[7] F.X. Deimbacher, Z.E. Heinemann, Time-dependent incorporation of locally irregular grids in large reservoir simulation models, in: SPE Symposium on Reservoir Simulation, Society of Petroleum Engineers, 1993.
[8] M.J. Mlacnik, A. Harrer, G. Heinemann, State-of-the-art in the windowing technique, in: Canadian International Petroleum Conference, Petroleum Society of Canada, 2001.
[9] I. Faille, F. Nataf, F. Willien, S. Wolf, Two local time stepping schemes for parabolic problems, in: ESAIM: Proceedings, vol. 29, EDP Sciences, 2009, pp. 58–72.
[10] W. Kheriji, R. Masson, A. Moncorgé, Nearwell local space and time refinement in reservoir simulation, Math. Comput. Simul. 118 (2015) 273–292, https://doi.org/10.1016/j.matcom.2014.11.022.
[11] G.W. Thomas, D.H. Thurnau, Reservoir simulation using an adaptive implicit method, Soc. Pet. Eng. J. 23 (1983) 759–768, https://doi.org/10.2118/10120-PA.
[12] P. Forsyth Jr, P. Sammon, Practical considerations for adaptive implicit methods in reservoir simulation, J. Comput. Phys. 62 (1986) 265–281, https://doi.org/10.1016/0021-9991(86)90127-0.
[13] T.F. Russel, Stability analysis and switching criteria for adaptive implicit methods based on the CFL condition, in: SPE Symposium on Reservoir Simulation, 1989.
[14] D.A. Collins, L.X. Nghiem, Y.K. Li, J.E. Grabonstotter, An efficient approach to adaptive-implicit compositional simulation with an equation of state, SPE Reserv. Eng. 7 (1992) 259–264, https://doi.org/10.2118/15133-PA.
[15] A. Moncorgé, H.A. Tchelepi, Stability criteria for thermal adaptive implicit compositional flows, SPE J. 14 (2009) 311–322, https://doi.org/10.2118/111610-PA.
[16] J. Maes, A. Moncorgé, H. Tchelepi, Thermal adaptive implicit method: time step selection, J. Pet. Sci. Eng. 106 (2013) 34–45, https://doi.org/10.1016/j.petrol.2013.03.019.
[17] H. Mc Namara, G. Bowen, P. Dellar, Locally adaptive timestepping in reservoir simulators, in: ECMOR XII-12th European Conference on the Mathematics of Oil Recovery, 2010.
[18] J.R. Appleyard, I.M. Cheshire, The Cascade method for accelerated convergence in implicit simulators, in: European Petroleum Conference, 25-28 October, London, United Kingdom, Society of Petroleum Engineers, 1982.
[19] J.W. Watts, A compositional formulation of the pressure and saturation equations, SPE Reserv. Eng. 1 (1986) 243–252, https://doi.org/10.2118/12244-PA.
[20] J.A. Trangenstein, J.B. Bell, Mathematical structure of the black-oil model for petroleum reservoir simulation, SIAM J. Appl. Math. 49 (1989) 749–783, https://doi.org/10.1137/0149044.
[21] O. Møyner, H.A. Tchelepi, A mass-conservative sequential implicit multiscale method for isothermal equation-of-state compositional problems, SPE J. 23 (2018) 2376–2393, https://doi.org/10.2118/182679-PA.
[22] A. Moncorgé, H.A. Tchelepi, P. Jenny, Sequential fully implicit formulation for compositional simulation using natural variables, J. Comput. Phys. 371 (2018) 690–711, https://doi.org/10.1016/j.jcp.2018.05.048.
[23] J.B. Bell, J.A. Trangenstein, G.R. Shubin, Conservation laws of mixed type describing three-phase flow in porous media, SIAM J. Appl. Math. 46 (1986) 1000–1017, https://doi.org/10.1137/0146059.
[24] K.-A. Lie, An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST), Cambridge University Press, 2019.
[25] L.D. Carciopolo, L. Formaggia, A. Scotti, H. Hajibeygi, Conservative multirate multiscale simulation of multiphase flow in heterogeneous porous media, J. Comput. Phys. 404 (2020) 109134, https://doi.org/10.1016/j.jcp.2019.109134.
[26] C.W. Gear, D.R. Wells, Multirate linear multistep methods, BIT Numer. Math. 24 (1984) 484–502, https://doi.org/10.1007/BF01934907.
[27] P. Jenny, Time adaptive conservative finite volume method, J. Comput. Phys. 403 (2020) 109067, https://doi.org/10.1016/j.jcp.2019.109067.
[28] F. Kwok, H. Tchelepi, Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media, J. Comput. Phys. 227 (2007) 706–727, https://doi.org/10.1016/j.jcp.2007.08.012.
[29] M. Shahvali, H.A. Tchelepi, Efficient coupling for non-linear multiphase flow with strong gravity, in: SPE Reservoir Simulation Symposium, 18-20 February, The Woodlands, Texas, USA, Society of Petroleum Engineers, 2013.
[30] J.R. Natvig, K.-A. Lie, B. Eikemo, I. Berre, An efficient discontinuous Galerkin method for advective transport in porous media, Adv. Water Resour. 30 (2007) 2424–2438, https://doi.org/10.1016/j.advwatres.2007.05.015.
[31] J.R. Natvig, K.-A. Lie, Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements, J. Comput. Phys. 227 (2008) 10108–10124, https://doi.org/10.1016/j.jcp.2008.08.024.
[32] K.-A. Lie, J.R. Natvig, H.M. Nilsen, Discussion of dynamics and operator splitting techniques for two-phase flow with gravity, Int. J. Numer. Anal. Model. 9 (2012) 684–700.
[33] K.-A. Lie, H.M. Nilsen, A.F. Rasmussen, X. Raynaud, Fast simulation of polymer injection in heavy-oil reservoirs on the basis of topological sorting and sequential splitting, SPE J. 19 (2014) 991–1004, https://doi.org/10.2118/163599-PA.
[34] Ø.S. Klemetsdal, A.F. Rasmussen, O. Møyner, K.-A. Lie, Efficient reordered nonlinear Gauss–Seidel solvers with higher order for black-oil models, Comput. Geosci. (2019), https://doi.org/10.1007/s10596-019-09844-5.
[35] A. Datta-Gupta, M.J. King, Streamline Simulation: Theory and Practice, SPE Textbook Series, vol. 11, Society of Petroleum Engineers, Richardson, TX, 2007.
[36] H. Cao, Development of techniques for general purpose simulators, Ph.D. thesis, Stanford University Stanford, CA, 2002.
[37] K. Aziz, A. Settari, Petroleum Reservoir Simulation, Applied Science Publishers, vol. 476, 1979.
[38] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, SPE Reserv. Eval. Eng. 4 (2001) 308–317, https://doi.org/10.2118/72469-PA, http://www.spe.org/csp/.

[39] A.N. Guion, B. Skaflestad, K.-A. Lie, X.-H. Wu, Validation of a non-uniform coarsening and upscaling framework, in: SPE Reservoir Simulation Conference, 10–11 April, Galveston, Texas, USA, Society of Petroleum Engineers, 2019.
[40] Ø.S. Klemetsdal, O. Møyner, K.-A. Lie, Implicit high-resolution compositional simulation with optimal ordering of unknowns and adaptive spatial refinement, in: SPE Reservoir Simulation Conference, 10-11 April, Galveston, Texas, USA, Society of Petroleum Engineers, 2019.
[41] Ø.S. Klemetsdal, K.-A. Lie, Dynamic coarsening and local reordered nonlinear solvers for simulating transport in porous media, SPE J. (2020), https://doi.org/10.2118/201089-PA (SPE-201089-PA), in press.