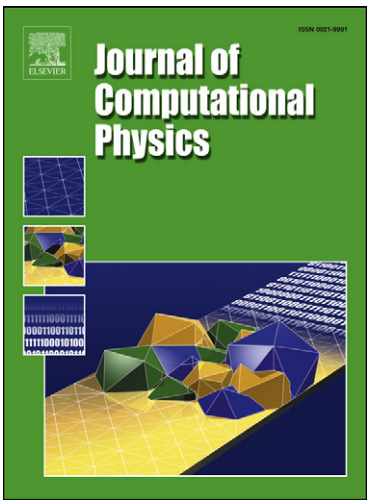


Local-global splitting for spatiotemporal-adaptive multiscale methods

Pavel Tomin, Ivan Lunati

PII: S0021-9991(14)00658-5
DOI: 10.1016/j.jcp.2014.09.022
Reference: YJCPH 5484

To appear in: *Journal of Computational Physics*

Received date: 21 February 2014
Revised date: 17 September 2014
Accepted date: 18 September 2014

Please cite this article in press as: P. Tomin, I. Lunati, Local-global splitting for spatiotemporal-adaptive multiscale methods, *J. Comput. Phys.* (2014), http://dx.doi.org/10.1016/j.jcp.2014.09.022

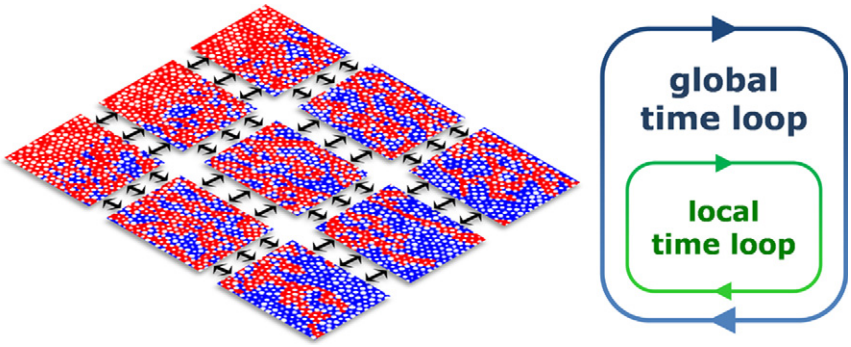
Graphical abstract

Journal of Computational Physics ●●●●, ●●●, ●●●

Local-global splitting for spatiotemporal-adaptive multiscale methods

Pavel Tomin*, Ivan Lunati

Institute of Earth Sciences (ISTE), University of Lausanne, Switzerland



Highlights

- Local-global splitting as a framework for multiscale multiphysics simulations.
- Spatiotemporal adaptivity for the Multiscale Finite Volume method.
- Adaptive criteria based on the changes of coarse-scale quantities.
- Self-correcting scheme to improve the quality of multiscale approximation.
- Multiscale multiphysics simulations of two-phase flow in porous media.

Local-global splitting for spatiotemporal-adaptive multiscale methods

Pavel Tomin*, Ivan Lunati

Institute of Earth Sciences (ISTE), University of Lausanne, Switzerland

Abstract

We present a novel spatiotemporal-adaptive Multiscale Finite Volume (MsFV) method, which is based on the natural idea that the global coarse-scale problem has longer characteristic time than the local fine-scale problems. As a consequence, the global problem can be solved with larger time steps than the local problems. In contrast to the pressure-transport splitting usually employed in the standard MsFV approach, we propose to start directly with a local-global splitting that allows to locally retain the original degree of coupling. This is crucial for highly non-linear systems or in the presence of physical instabilities. To obtain an accurate and efficient algorithm, we devise new adaptive criteria for global update that are based on changes of coarse-scale quantities rather than on fine-scale quantities, as it is routinely done before in the adaptive MsFV method. By means of a complexity analysis we show that the adaptive approach gives a noticeable speed-up with respect to the standard MsFV algorithm. In particular, it is efficient in case of large upscaling factors, which is important for multiphysics problems. Based on the observation that local time stepping acts as a smoother, we devise a self-correcting algorithm which incorporates the information from previous times to improve the quality of the multiscale approximation. We present results of multiphase flow simulations both for Darcy-scale and multiphysics (hybrid) problems, in which a local pore-scale description is combined with a global Darcy-like description. The novel spatiotemporal-adaptive multiscale method based on the local-global splitting is not limited to porous media flow problems, but it can be extended to any system described by a set of conservation equations.

Keywords: multiscale methods, spatiotemporal adaptivity, local-global splitting, multiphysics

1. Introduction

Multiphase flow and transport in porous media are characterized by multiple spatiotemporal scales, that span several orders of magnitude. Spatial scales, for instance, range from pore size (fractions of a millimeter or less) to field or regional scale (tens of kilometers or more) [1, 2]. In general, such a disparity of scales suggests that a macroscopic description can be obtained without resolving the microscopic details. In

*Corresponding author
Email addresses: pavel.tomin@unil.ch (Pavel Tomin), ivan.lunati@unil.ch (Ivan Lunati)

geological porous media, however, the macroscopic and microscopic scales are coupled by the presence of long-correlation pathways that originate either from a heterogeneity of the formation or as a result of non-linear processes (e.g., multiphase flow and reactive transport) that give raise to instabilities and lead to the emergence of new characteristic lengths. Under these circumstances, it is necessary to resolve the fine-scale details because they impact the macroscopic description of the system. However, a direct description of the smallest relevant scales remains computationally very challenging due to the enormous number of degrees of freedom required.

Multiscale algorithms has been developed to avoid this computational bottleneck and handle the increasing number of unknowns accurately and efficiently. The basic idea of multiscale algorithms is to split the original problem into a set of local fine-scale sub-problems coupled by a global coarse-scale problem. Local problems are solved to provide a numerical closure to the global problem (by means of interpolators which are local numerical solutions) and to reconstruct the fine-scale details of the solution.

The Multiscale Finite Volume (MsFV) method was proposed originally as a solver for elliptic problems describing the pressure distribution in an incompressible fluid flowing through a porous medium [3]. Later it has been extended to deal with physically complex processes such as parabolic problems arising in compressible multiphase flow [4, 5], gravity currents [6] and density-driven instabilities [7, 8], three-phase flow with solution gas [9], compositional processes [10], flow in presence of complex wells [11, 12], flow in fractured porous media [13, 14], elliptic problem in a predictor-corrector method to solve Navier-Stokes equations [15]. Recently, the MsFV method has been proposed as a general conservation-based framework to couple multiple scales in multiphysics applications that employ the Navier-Stokes equations at the microscale and Darcy's law at the macroscale [16].

All these problems involve, in addition to a pressure equation, the solution of a transport equation, which requires the definition of a velocity field. In solving these time-dependent problems, the MsFV method relies on a pressure-transport splitting. After the equations are split, the variants of original MsFV formulation are basically applied to the pressure equation (elliptic or parabolic), whereas the transport equation is treated separately and differently in virtue of the dominant role played by advection. This has fostered the development of strategies that adaptively update the interpolators in case of coupled non-linear problems [17, 18, 19] or adaptively refine the solution in a front region [20, 7, 8]. All these strategies implement the spatial adaptivity, but the global solution is computed at every time. This is unnecessary because, as a results of the larger spatial scale, the global problem is also characterized by a longer characteristic time and can be solved with larger time steps than the local problems. Motivated by this simple observation we propose a solution strategy that is adaptive in space and time and employs different time steps for global and local problems. To the best of our knowledge, the differences in time scales was never considered before in the context of the MsFV method. Moreover, the local-global formulation increases the flexibility of the algorithm and allows us to construct a general conservation-based framework which is not limited to elliptic problems but it can

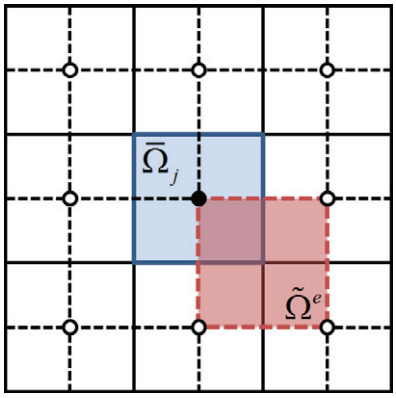


Figure 1: Coarse (solid) and dual (dashed) grids. Underlying fine grid not shown here.

be extended to an arbitrary set of balance equations.

The paper is organized as follows. We start with a brief description of the MsFV method using the operator formulation of the algorithm; then we propose a local-global splitting of the system of equations showing applications to Darcy flow and to multiphysics algorithms (Section 2). Next, we show the implication of this local-global splitting that allows spatiotemporal adaptivity and a self-correcting solution scheme (Section 3). Then, we present some numerical examples to illustrate the performance of the proposed strategies (Section 4), and we discuss the results (Section 5).

2. The MsFV method with local-global splitting

The MsFV algorithm is based on an auxiliary finite-volume discretization at the coarse scale and requires coarse control volumes, global degrees of freedom, and interpolators. This entails the construction of two auxiliary coarse grids (Fig. 1): a primal coarse grid, which defines the coarse control volumes, $\bar{\Omega}_j$; and a dual coarse grid, which defines the subdomains, $\tilde{\Omega}^e$, on which the interpolators are computed as solutions of a set of localized numerical problems [3, 6]. The interpolators are used to compute the fluxes across the control volumes and to construct the problem for the global degrees of freedom, which are associated with the nodes of the dual grid. Once the global problem is solved, the interpolators may be used to prolongate the coarse solution back to the original fine-scale grid.

In general, the MsFV method is employed in problems which require computing a velocity field to be used in a transport equation (e.g., for the problems describing multiphase flow in porous media, see Appendix A for the governing equations). As the interpolators are computed locally, the solution obtained by their linear combination is not conservative across the boundaries of dual cells. To avoid balance errors, the interpolated solution is not used in the transport equation. Instead, it is employed to calculate the fluxes across the boundary of the coarse cells; then, these fluxes are assigned as boundary conditions to solve

localized problems on coarse cells and obtain an approximate but conservative flux field, which is used in the transport equation without introducing balance errors [4, 6].

Algorithm 1 Standard MsFV algorithm (see [21] for the details)

0. **Preprocessing**: construct the auxiliary coarse grids (primal and dual)

repeat the time loop

pressure:
1. **Interpolators**: construct basis and correction functions (interpolators)
2. **Coarse problem**: compute the coarse solution
3. **Flow B.C.**: construct the flux boundary conditions by interpolation
4. **Flow solution**: solve the local problems to obtain the conservative fluxes

saturation:
5. **Transport B.C.**: construct the transport boundary conditions
6. **Transport solution**: solve the local transport problems

until the end of simulation

The outline of the standard MsFV algorithm is given in Alg. 1 (we refer the reader to [21] for a comprehensive description of the method). After the auxiliary coarse grids are constructed (step 0), the original system of equations is first split into a pressure equation and a transport equation; then the MsFV is applied to the pressure system to obtain a conservative flux field (steps 1–4); finally, the transport is solved separately (steps 5–6) [17, 18]. Although some procedures have been proposed to better integrate the transport equation [20, 7], the original system is always first split into a pressure and a transport part, which are solved sequentially, and then into global and local problems. In the following we give a slightly different presentation of the method and we first introduce a local-global splitting, which locally conserves the original coupling between the equations and naturally leads us to introduce different time steps for the local and global problems. The flowchart of new algorithm is presented in Fig. 2.

2.1. Local problems

We consider the discrete form of a system of (conservation) equations, i.e.,

$$Ax = r, \tag{1}$$

where x is the vector of the unknowns; A is the coefficient matrix, which arises from the discretization of a linear (or linearized) problem; and r is the right-hand-side vector (this term contains the non-homogeneous part of the equation, and may include source terms, the effects of the boundary conditions, or the terms arising from the linearization). In principle, Eqs. 1 may represent any system of conservation equations

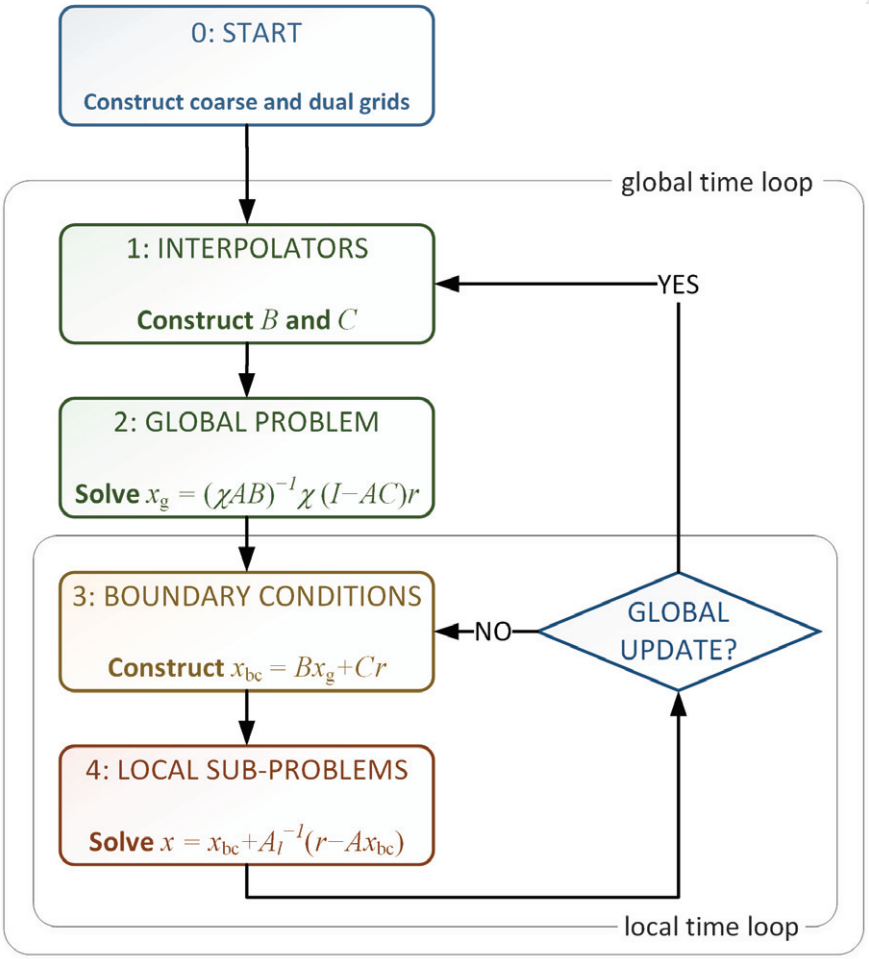


Figure 2: Flowchart of the multiscale algorithm based on the local-global splitting. First, the auxiliary coarse and dual grids are constructed (step 0). Then, the interpolators are computed locally in the cells of dual grid (step 1) and the global problem is constructed (step 2). These two steps form the global part of the algorithm. After the global problem has been solved, the boundary conditions for the local sub-problems are constructed (step 3) and the full system of equations is solved locally (step 4) until a global update is required. Notice that x_g changes only after step 2, thereby only the last term in equation for x_{bc} (step 3) is modified in the local time loop. This term, in particular, contains the values of variables for which global degrees of freedom are not defined by virtue of the hyperbolic nature of their governing equation.

(mass, momentum, or energy balance). In Appendix A we present the governing equations of Darcy-scale and pore-scale flow trough porous media, which are the applications considered here; but the approach is general and not limited to these test cases.

The main idea is to spatially decompose Eqs. 1 into a set of local problems defined on subdomains, $\bar{\Omega}_j$, which are coupled through the boundary conditions. The local sub-problems can be written as

$$A_\ell x = r + (A_\ell - A)x_{\rm bc}, \tag{2}$$

or in the equivalent residual form as

$$x = x_{\rm bc} + A_\ell^{-1}(r - Ax_{\rm bc}), \tag{3}$$

where $x_{\rm bc}$ is an approximate solution which provides the boundary conditions, and A_ℓ is the operator that defines the set of localized problems on coarse cells and is constructed from the exact fine-scale operator A by removing the connections across coarse cells (it has a block-diagonal structure if the unknowns are appropriately ordered, see [21] for details).

Notice that, in contrast to previous applications of the MsFV method, the local sub-problems retain the original level of coupling between the equations as the full system of equations is solved locally (step 4 in Fig. 2). Moreover, different physics and/or numerical schemes (e.g. a fully coupled implicit scheme) may be used for local sub-problems.

Here, we have simply decomposed the original problem into a set of local sub-problems. In the next section we describe the construction of the boundary conditions, which couple the sub-problems.

2.2. Global problem and boundary conditions

The boundary conditions assigned to the local problems describe the effects of global quantities. They are obtained by solving a global problem, which is constructed by projecting the fine-scale problem onto the coarse grid using a restriction operator χ, which in the MsFV method is the control volume summation operator that sums up all fine-scale values corresponding to the same coarse cell (this procedure guarantees coarse-scale conservation). The resulting global problem is

$$\chi A x_{\rm bc} = \chi r, \tag{4}$$

and its solution is assumed to have the following form

$$x_{\rm bc} = \mathcal{B} x_{\rm g} + \mathcal{C} r, \tag{5}$$

where the columns of the basis-function operator, $\mathcal{B}$, contain the interpolators and provide a numerical closure to the global problem. The correction function operator, $\mathcal{C}$, accounts for the local effects of the

right-hand side, r, [6, 21]. Both operators are calculated on staggered supports defined by the dual cells, $\tilde{\Omega}_e$, (step 1 in Fig. 2).

Substituting this decomposition, Eq. 5, back into Eq. 4, we obtain the global problem

$$(\chi A\mathcal{B})x_{\mathrm{g}} = \chi(I - A\mathcal{C})r, \tag{6}$$

which can be solved for the global degrees of freedom, x_{g}, (step 2 in Fig. 2). After the global solution has been computed, the boundary conditions for local problems are obtained by interpolation using again Eq. 5 (step 3 in Fig. 2).

In principle, global degrees of freedom could be defined for all variables and the interpolators could be solutions of the full systems of equations with the original level of coupling. In practice, however, we can choose the interpolators and the variables for which we need to define global degrees of freedom based on the nature of the corresponding equation: elliptic and parabolic equations would require the definition of a global problem because of long-scale interaction, whereas hyperbolic equations might be efficiently solved without defining a corresponding global problem since the interaction is local.

In the following, we focus on multiphase flow problems for which the unknown vector is $x = [\ p \quad u \quad s\ ]^T$ (p is the pressure, u the (total) velocity, and s the phase indicator) and we consider two cases: a Darcy-scale flow problem, which represents a classic MsFV application, and a pore-scale problem, in which a selective choice of the global degrees of freedom leads to a multiphysics algorithm that solves Navier-Stokes equations with interfaces in pores, but uses a Darcy-like description for the global problem at the coarse scale. For both cases, the full set of governing equations is presented in Appendix A; and in the following we focus on the form of the discretized system. Notice that in the Darcy-scale problem the interpolators are solutions of a single equation for pressure; whereas in the pore-scale problem they are solutions of a system of equations for u and p [16].

2.3. Darcy-scale multiphase flow

We consider the flow of two incompressible phases in a rigid porous medium. (The governing equations are presented in Appendix A.1.) We write the system in Eqs. 1 in the form

$$Ax = \begin{bmatrix} \mathcal{G}^T\mathcal{D} & 0 & 0 \\ -\mathcal{D} & 1 & 0 \\ 0 & \mathcal{A} & \mathcal{T} \end{bmatrix} \begin{bmatrix} p \\ u \\ s \end{bmatrix} = \begin{bmatrix} r_p \\ r_u \\ r_s \end{bmatrix} = r, \tag{7}$$

where $\mathcal{D} = -\Lambda\mathcal{G}$ is the Darcy operator that maps the pressure field into the velocity field, Λ is a diagonal matrix containing the values of the total mobility at the fine-cell interfaces; $\mathcal{G}$ is the gradient operator and its transposed, $\mathcal{G}^T$, is the divergence operator; $\mathcal{A}$ and $\mathcal{T}$ are two operators related to the transport equation: the former represents the dependence of the advection term on the velocity.

In Eq. 7, the first row represents the elliptic equation (A.3) which is solved for the pressure ($p = (\mathcal{G}^T\mathcal{D})^{-1}r_p$); the second row is Darcy's law (A.2), which allows computing the velocity from the pressure field ($u = \mathcal{D}p + r_u = \mathcal{D}(\mathcal{G}^T\mathcal{D})^{-1}r_p + r_u$); and the third row is the hyperbolic transport problem (A.4) (here, for simplicity, capillary pressure is neglected).

Based on the different nature of pressure and saturation equations, we define the approximate solution (5) used to compute the boundary conditions as

$$x_{\mathrm{bc}} = \begin{bmatrix} p_{\mathrm{bc}} \\ u_{\mathrm{bc}} \\ s_{\mathrm{bc}} \end{bmatrix} = \begin{bmatrix} \mathcal{B}_p \\ \mathcal{B}_u \\ 0 \end{bmatrix} p_{\mathrm{g}} + \mathcal{C}r, \tag{8}$$

where we have assumed that the global pressures, p_{g}, defined at the centers of the coarse cells (which are the nodes of the dual grid) are the only global degrees of freedom. According to Darcy's law, the relationship $\mathcal{B}_u = \mathcal{D}\mathcal{B}_p$ holds between velocity and pressure basis functions ($\mathcal{B}_u$ and $\mathcal{B}_p$, respectively), which are therefore not independent. Since the saturation equation is hyperbolic, the boundary conditions for the localized saturation problems, s_{bc}, are not computed by means of interpolators ($\mathcal{B}_s = 0$) but they are simply taken from the saturation field at previous time or iteration (depending on the coupling scheme employed for the saturation equation) and constructed by applying the correction-function operator to the right-hand side.

The pressure basis functions are constructed as extensions of bilinear interpolators and represent the effects of a unit pressure at the nodes of the dual grid on the coarse cell balances (see, e.g., [21] for the details about the basis function operator). The global pressure is solution of

$$(\chi_p A\mathcal{B})p_{\mathrm{g}} = \chi_p(I - A\mathcal{C})r, \tag{9}$$

where we have added a subscript to the summation operator, χ_p, to indicate that only a global equation for the pressure is constructed (with abuse of notation on the use of the summation operator we can write the global operator as $(\chi_p\mathcal{G}^T\mathcal{B}_u) = (\chi_p\mathcal{G}^T\mathcal{D}\mathcal{B}_p)$). Once the global problem (9) is solved, the boundary conditions for the full system of equations can be constructed from Eq. 8 and used to solve Eqs. 2 locally. Mass conservation is guaranteed by the use of a conservative total-flux field, which results from the consistency between global and local fluxes; whereas updating the saturation at every time or iteration step ensures a correct phase transport across subdomain boundaries, with a saturation that remains bounded between 0 and 1.

2.4. Multiphysics algorithm: Darcy- and pore-scale descriptions

In addition to the classic application of the MsFV method to Darcy-scale flow, we consider the pore-scale description of multiphase flow. In this case, the Navier-Stokes equations have to be solved in the pore

geometry to compute the velocity field. To track the evolution of the fluid-fluid interface we use the Volume Of Fluid method (VOF), which is based on a whole domain formulation of the problem [22, 23, 24, 25, 26, 27]. (The governing equations are presented in Appendix A.2.)

In this case, the system in Eqs. 1 can be represented in the form

$$Ax = \begin{bmatrix} 0 & \mathcal{G}^T & 0 \\ -\mathcal{G} & \mathcal{N} & 0 \\ 0 & \mathcal{A} & \mathcal{T} \end{bmatrix} \begin{bmatrix} p \\ u \\ s \end{bmatrix} = \begin{bmatrix} r_p \\ r_u \\ r_s \end{bmatrix} = r, \tag{10}$$

where the first row represents the divergence-free condition (A.7) on the velocity, which is derived from the total mass balance for two incompressible phases (r_p contains the global boundary conditions); the second row contains the Navier-Stokes equations (A.8) for momentum balance ($\mathcal{N}$ represents the operator acting on the velocity and r_u includes the effects of the surface forces); and the third row is the advection equation (A.9) for the fluid indicator.

The MsFV method can be applied to this system of equations to obtain a multiphysics algorithm that couples a fine-scale description based on the Navier-Stokes equations with a Darcy-like description of the flow at the global scale. This can be done simply by assuming that the global pressures are the only relevant degrees of freedom at the global scale [16]. Therefore, the approximate solution used to compute the boundary conditions is again Eq. 8, but now pressure and velocity interpolators are no longer related through Darcy's law ($\mathcal{B}_u \neq \mathcal{D}\mathcal{B}_p$) and are computed by simultaneous solutions of pressure and velocity equations for each degrees of freedom (see [16] for a comprehensive description of basis and correction functions). The global-pressure problem is again given by Eq. 9. The elements of the global-scale operator (which, again with abuse of notation on the use of the summation operator, we can write as $(\chi_p \mathcal{G}^T \mathcal{B}_u)$) can be interpreted as an extension of Darcy-scale total mobility in the sense that they do not simply depend on the average saturation, but on the actual spatial distribution of the two fluids [16].

Once the global pressure problem is computed, the full system of pore-scale equations is solved locally on coarse cells, $\bar{\Omega}_j$, with boundary conditions obtained from Eq. 8 (again, the boundary conditions for s are taken from the fluid indicator field at the previous time). Mass conservation is ensured by the consistency between the integral pore-scale fluxes across the boundary, $\partial\bar{\Omega}_j$, and the corresponding Darcy-scale fluxes; whereas the solvability of the local momentum-balance equations (for which no global balance is imposed) is guaranteed by the fact that the drag exerted by the solid obstacles acts as a momentum sink.

3. Spatiotemporal adaptivity and self-correcting solution

Splitting the original problem into the set of local sub-problems Eqs. 2 coupled together by the global problem Eq. 6 makes the algorithm more flexible and opens new possibilities in terms of spatiotemporal adaptivity because two temporal scales (global and local) are naturally inherited from the two spatial scales.

As the local problems retain the original degree of coupling, the full system of equations can be solved locally with smaller time steps, Δt_l, than the time step used to solve the global problem, ΔT_g. This means that global solution, x_{g}, and corresponding part of local boundary conditions (see Eq. 5 for x_{bc}) are updated only every $G = \Delta T_g / \Delta t_l \geq 1$ local time steps, and this value can be chosen adaptively. In addition to reducing the computational costs, the tighter coupling, achieved by locally solving the full system of equations, may improve the accuracy and the stability of the MsFV approach, which, based on our experience, is usually less stable than the original fine-scale problem. In the following, we propose several adaptivity criteria to selectively solve the global problem and we discuss the complexity of the method. Also, we show that spatiotemporal adaptivity allows constructing a self-correcting algorithm which improves the accuracy of the solution by incorporating information from previous times.

3.1. Adaptive criteria for local-global time stepping

In the MsFV framework, standard adaptive strategies are based on total-mobility variation in fine cells [17, 18, 19, 20]. This is justified when the ratio between the characteristic fine cell size, h_f, and the coarse cell size, H_c, is of order one, $\epsilon = h_f / H_c \lesssim 1$, and local changes significantly influence the global degrees of freedom. However, if $\epsilon \ll 1$, the global effects of changes in fine cells might be negligible and global update criteria should be based on coarse-scale quantities.

An extension of the mobility-based criterion to the coarse scale is not computationally efficient because it requires an update of the basis function operator, which contains the coarse-scale transmissibilities. Here, we propose two criteria that aim at an indirect estimate of coarse-scale mobility changes: a Courant-Friedrichs-Levy (CFL) criterion and a pressure-gradient criterion.

CFL criterion

The CFL criterion is based on the assumption that the global solution is affected only by local saturation changes over lengths larger than a characteristic value, ℓ, which is $h_f \ll \ell \sim H_c$. The definition of the threshold length ℓ naturally leads to a Courant-Friedrichs-Levy (CFL) condition

$$\frac{U_c \Delta T_g}{H_c} \leq \mathrm{C_c} = \frac{\ell}{H_c} \approx \delta_{\mathrm{CFL}} \cdot \mathrm{C_f}, \tag{11}$$

where U_c is the coarse velocity, and δ_{CFL} is a parameter relating the coarse-scale CFL number, $\mathrm{C_c}$, to the fine-scale CFL number, $\mathrm{C_f}$. δ_{CFL} can be treated as a free parameter which determines the accuracy and the efficiency of the solution for advection-dominated processes: larger values favor efficiency, whereas smaller values produce more accurate solutions.

Assuming that the characteristic velocity is scale-independent as in [28] leads to the following relationship between global and local time steps

$$\Delta T_g = \delta_{\mathrm{CFL}} \Delta t_l \frac{H_c}{h_f}. \tag{12}$$

Here, in analogy to [29], we account for the possibility that fine-scale velocity, u_f, and the coarse-scale velocity, U_c, may be different, which leads to

$$\Delta T_g = \delta_{\mathrm{CFL}} \Delta t_l \frac{H_c}{h_f} \frac{u_f}{U_c}, \tag{13}$$

where the relationships between u_f and U_c can be chosen accordingly to the problem considered and to the variable of interest. A distinction between u_f and U_c is particularly important in case of very heterogeneous velocity field (as in presence of channelized flow). We assume that $U_c = \max_j \langle u \rangle_{\bar{\Omega}_j}$.

Notice that, although the CFL criterion seems reasonable for transport, it does not explicitly account for the coupling between flow and transport processes. This might lead to contradictions when, for instance, two fluids with identical properties are considered: the global pressure field has to be updated in accordance with the CFL condition, whereas the transport of the fluids has no effect on the velocity. Therefore this criterion should be complemented (or substituted) by a flow-based condition.

Pressure-gradient criterion

The pressure-gradient criterion is based on the observation that, when the fluxes across the coarse cell boundary are fixed, the average velocity in the enclosed volume is constant. Therefore, changes in the pressure force applied at the boundary indicate changes in the global flow properties due to fluid redistribution in the cell. Defining the effective mobility of the volume, $\lambda_{\bar{\Omega}}$, as the coefficient of proportionality between the average velocity and the average pressure force,

$$\langle \mathbf{u} \rangle_{\bar{\Omega}} = \lambda_{\bar{\Omega}} \langle \nabla p \rangle_{\bar{\Omega}}, \tag{14}$$

we have

$$0 = \Delta \langle \mathbf{u} \rangle_{\bar{\Omega}} = \Delta \lambda_{\bar{\Omega}} \langle \nabla p \rangle_{\bar{\Omega}} + \lambda_{\bar{\Omega}} \Delta \langle \nabla p \rangle_{\bar{\Omega}}, \tag{15}$$

Then, the relative variation of the mobility can be approximated by the relative variation of the applied pressure force

$$\left| \frac{\Delta \lambda_{\bar{\Omega}}}{\lambda_{\bar{\Omega}}} \right| \approx \frac{|\Delta \langle \nabla p \rangle_{\bar{\Omega}}|}{|\langle \nabla p \rangle_{\bar{\Omega}}|}, \tag{16}$$

and the average pressure gradient in $\bar{\Omega}$ can be expressed as the fine-scale pressure force applied at the coarse cell boundary per unit volume,

$$\langle \nabla p \rangle_{\bar{\Omega}} = \frac{1}{V_{\bar{\Omega}}} \int_{\bar{\Omega}} \nabla p \, dV = \frac{1}{V_{\bar{\Omega}}} \oint_{\partial \bar{\Omega}} p \, \boldsymbol{n} \, dS. \tag{17}$$

If the relative change in applied pressure force does not exceed a certain threshold value, i.e.,

$$\frac{|\Delta \langle \nabla p \rangle_{\bar{\Omega}}|}{|\langle \nabla p \rangle_{\bar{\Omega}}|} = \frac{\left| \Delta \oint_{\partial \bar{\Omega}} p \, \boldsymbol{n} \, dS \right|}{\left| \oint_{\partial \bar{\Omega}} p \, \boldsymbol{n} \, dS \right|} \lesssim \delta_{\nabla p}, \tag{18}$$

the effective mobility of the volume does not sensitively change; therefore, the global solution is updated only when the threshold value, $\delta_{\nabla p}$, is overcome in at least one of the coarse cells.

3.2. Complexity analysis

To simplify the analysis, we consider only the case of Darcy-scale flow for which the costs are dominated by the solution of the pressure equation. We assume that the cost of solving a linear system with N unknowns is $\zeta_L = aN^\alpha$, where the prefactor a and the exponent α depends on the specific solver. For a direct solver, we can assume $a = 1$ and $\alpha = 2$; whereas for an iterative solver we might have $a \approx 50$ and $\alpha \approx 1.2$ (which corresponds to an optimal multigrid solver [30]). Following [31], the costs of the MsFV method can be estimated as

$$\zeta_{\mathrm{MS}} = [nFN_c a\Upsilon^\alpha + aN_c^\alpha] + aFN_c\Upsilon^\alpha, \tag{19}$$

where Υ is the upscaling factor which is related to the scale ratio, $\Upsilon \approx \epsilon^{-d} = (H_c/h_f)^d$ (d is the dimension); $N_c = N_f/\Upsilon$ is the number of coarse or dual cells (which are assumed to be approximately the same since N_c is large); and N_f is the number of fine cells. The quantity in brackets refers to the construction of the boundary conditions for the local problems: the first term estimates the cost of constructing n basis and correction functions, and $F \leq 1$ is adaptivity factor, which expresses the fraction of the interpolators that is updated at each time step or iteration [17, 18, 20, 19, 7]; the second term is the cost of the global problem. The last term on the right-hand side refers to the solution of local problems. Notice that we have applied the same adaptivity factor F to the cost of the local problems as they can be solved only in the regions of interest [20, 7, 8]. If an iterative solver is used $n = (2^d + 1)$. However, when direct solvers are employed, it is possible to store the factorization of the operator that can be reused for all basis and correction functions; in this case n can be omitted (or equivalently we can assume $n \approx 1$) as the cost of the factorization dominates the cost of computing the solutions. Notice that if the local problems are too large (e.g., $\Upsilon \gtrsim 10^5$ as in case of multiphysics simulations), the direct solver is not applicable and iterative solver should be used.

The local-global splitting allows us to adaptively update the boundary conditions, which reduces the cost by a factor $G^{-1} \leq 1$. Using the relationship $N_c = N_f/\Upsilon$, we write

$$\zeta_{\mathrm{LG}} = aG^{-1}\left(nFN_f\Upsilon^{\alpha-1} + N_f^\alpha\Upsilon^{-\alpha}\right) + aFN_f\Upsilon^{\alpha-1}, \tag{20}$$

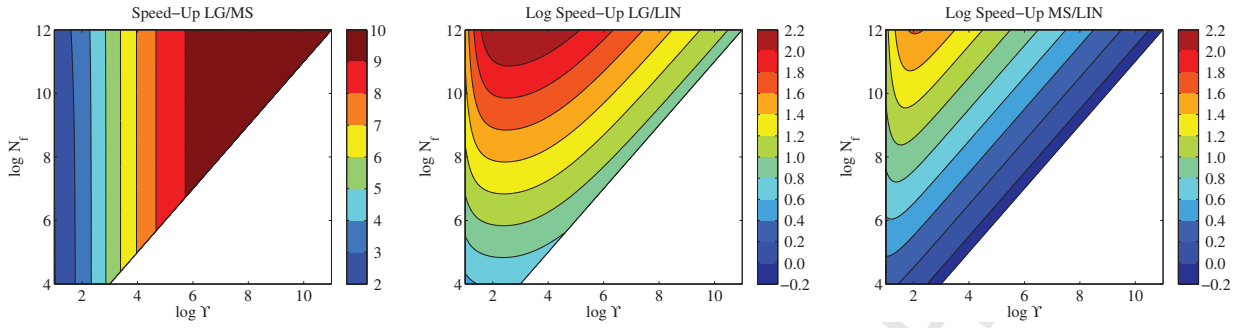
where the coefficient G^{-1} depends on the specific adaptive criterion and tolerance that are employed. For $G = 1$ we recover the costs of the MsFV method.

The speed-up offered by the spatiotemporal adaptivity with respect to the MsFV is

$$\frac{\zeta_{\mathrm{MS}}}{\zeta_{\mathrm{LG}}} = \frac{\left(n + F^{-1}N_f^{\alpha-1}\Upsilon^{-2\alpha+1}\right) + 1}{G^{-1}\left(n + F^{-1}N_f^{\alpha-1}\Upsilon^{-2\alpha+1}\right) + 1}, \tag{21}$$

whereas the speed-up with respect to the fine-scale solver is

$$\frac{\zeta_{\mathrm{L}}}{\zeta_{\mathrm{LG}}} = \frac{1}{G^{-1}\left(nFN_f^{1-\alpha}\Upsilon^{\alpha-1} + \Upsilon^{-\alpha}\right) + FN_f^{1-\alpha}\Upsilon^{\alpha-1}}. \tag{22}$$



(a) (b) (c)

Figure 3: Comparison between the costs of the standard MsFV method, of the spatiotemporal-adaptive approach, and of the fine-scale solver: (a) speed-up of the spatiotemporal-adaptive algorithm with respect to the standard MsFV method; (b) common logarithm of the speed-up of the spatiotemporal-adaptive algorithm with respect to the fine-scale solver; (c) common logarithm of the speed-up of the standard MsFV method with respect to the fine-scale solver. We consider a three-dimensional problem, $d = 3$, an iterative solver ($a = 50$, $\alpha = 1.2$, $n = 9$), and the adaptivity factor $F = 0.2$.

The comparison between the costs of the standard MsFV method and the spatiotemporal-adaptive algorithm is presented in Fig. 3a together with the comparison of the multiscale approaches with fine-scale solver (Fig. 3b and 3c). We have assumed that an iterative solver is used ($a = 50$, $\alpha = 1.2$, $n = 9$) to solve a three-dimensional problem ($d = 3$) and that the adaptivity factor is $F = 0.2$ (see, e.g., [20]). To estimate G we have used a simple criterion based on Eq. 12 with $\delta_{\mathrm{CFL}} = 1$, which yields $G = \Upsilon^{1/d}$. The spatiotemporal-adaptive algorithm is always faster than the standard MsFV (Fig. 3a) and for the limiting case $G \gg 1$ and $\Upsilon \gg 1$ the speed-up is $n + 1$, which gives $\zeta_{\mathrm{MS}}/\zeta_{\mathrm{LG}} = 10$ for 3D problems. In comparison to the fine-scale solver, the spatiotemporal adaptivity leads to a faster algorithm for the whole range of parameters (Fig. 3b), while the MsFV is slower for the largest values of Υ (at given N_f, Fig. 3c). Also, a reasonable speed-up is achieved for a much wider region of the parameter space (compare Fig. 3b and 3c), making it possible to choose the upscaling parameter Υ on the basis of accuracy considerations (e.g., for multiphysics applications [16]).

3.3. Self-correcting solution

The quality of multiscale solution is directly related to the quality of the interpolators (basis and correction functions) used to construct the global problem and to prolongate the solution back to the fine grid. In the MsFV method, the interpolators are computed by solving a set of problems localized on the dual cells [18, 6, 21]. The localization is achieved by neglecting the transversal fluxes across the boundary of the dual cells [3]. This is usually a good approximation if the heterogeneity scale is much smaller than coarse cell size (scale-separation hypothesis) or if the dual cell boundaries coincide with highly conductive channels (e.g., fractures [14]). In other situations, this simple localization may be quite inaccurate and several strate-

gies have been proposed to iteratively improve the boundary conditions for basis and correction functions by estimating the transversal fluxes from some previous solution [32, 33, 34]. Using the definition of the multiscale operator, $\mathcal{M} = [\mathcal{B}(\chi A\mathcal{B})^{-1}\chi(I - A\mathcal{C}) + \mathcal{C}]$ (I is the identity matrix), this can be written as

$$x^{\mu}_{\mathrm{bc}} = \mathcal{M}^{-1}r + \left(I - \mathcal{M}^{-1}A\right)x^{\nu}_{\mathrm{bc}} = x^{\nu}_{\mathrm{bc}} + \mathcal{M}^{-1}(r - Ax^{\nu}_{\mathrm{bc}}), \tag{23}$$

where the index μ denotes the current iteration level, and ν the previous solution, x^{ν}_{bc}, from which the fluxes across the boundary are estimated, $\left(I - \mathcal{M}^{-1}A\right)x^{\nu}_{\mathrm{bc}}$. Using the solution at the previous iteration (i.e. $x^{\nu}_{\mathrm{bc}} = x^{\mu-1}_{\mathrm{bc}}$) would lead to an unstable scheme and some stabilization procedures has to be applied, such as the Generalized Minimal Residual (GMRES) method [33] or smoothing [32]. Notice that, in contrast to the GMRES-based approach that automatically guarantees a non-divergent scheme [35, 36], a sufficiently large number of steps is required if a smoother is employed.

In case of time-dependent problems, instead of iterating, we can use the local solution at the previous time to estimate the fluxes across the dual cell boundaries. The self-correcting scheme can be written in a form similar to Eq. 23,

$$x^{T}_{\mathrm{bc}} = \mathcal{M}^{-1}r + \left(I - \mathcal{M}^{-1}A\right)x^{t} = x^{t} + \mathcal{M}^{-1}(r - Ax^{t}), \tag{24}$$

where x^{T}_{bc} denotes the solution (used to compute the boundary condition of the local problem on coarse cells) at the new time step, and x^{t} is the local solution at the previous time step. (The subscript T and t denote global and local time steps, respectively.) Analogously to Eq. 23, this scheme will be stable if x^{t} is smooth enough, that is if a sufficient number of local time steps are used between two global updates (i.e., $G \gg 1$) because the local time stepping will act as a smoother. Notice that in this manner, the quality of the multiscale solution is improved at no additional computation cost by using information from the local solution, which is usually neglected in the standard MsFV method.

However, similar to the iterative scheme in Eq. 23, the self-correcting scheme is only conditionally stable and a minimum number of local time steps should be employed to stabilize the solution. As this number is in general not known a-priori, we apply a criterion based on the detection of an exponential growth of the magnitude of the flux correction, $c^{T} = \left\|\left(I - \mathcal{M}^{-1}A\right)x^{t}\right\|$. If the growth is faster than exponential with time,

$$\frac{c^{T_1} - c^{T_2}}{c^{T_2} - c^{T_3}} > \frac{\exp(T_1 - T_2) - 1}{1 - \exp\left(T_3 - T_2\right)}, \qquad T_1 > T_2 > T_3, \tag{25}$$

flux are not corrected for the current time. This allows stabilizing the correction scheme and avoiding divergence of the solution.

4. Numerical results

To test the advantages offered by the spatiotemporal-adaptive scheme, we consider a set of typical problems. First, we illustrate the performance of spatiotemporal adaptivity and self-correction in terms of

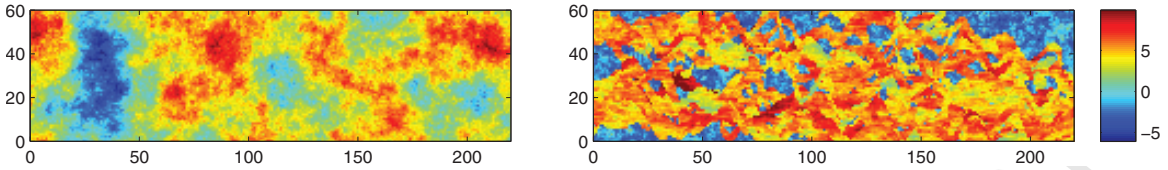


Figure 4: Logarithm of permeability fields of SPE10 top (left) and SPE10 bottom (right) layers.

accuracy and efficiency for Darcy-scale flow problems in heterogeneous media; then, we present a multiphysics application.

4.1. Darcy-scale flow in heterogeneous formations

We consider a two-dimensional rectangular domain of size 220 m by 60 m, which is discretized into 220×60 fine cells. The coarse grid contains 44×12 cells, which corresponds to an upscaling factor $\Upsilon = 5^2$. A fixed incoming flux is specified at the left boundary while a constant pressure is assigned at the right boundary. The top and the bottom boundaries are impermeable. We use quadratic relative permeabilities, $k_{r\beta} = s_\beta^2$; an unfavorable viscosity ratio $M = \mu_1/\mu_2 = 0.1$; and a constant porosity $\phi = 0.1$. Locally we employ a sequential implicit scheme [18] (alternatively, a fully implicit approach may be employed), and for the adaptive simulations we solve the global pressure system only at selected times, making global solution scheme similar to the IMplicit Pressure Explicit Saturation (IMPES) algorithm [37].

To study the performance of the spatiotemporal-adaptive strategy, we consider two heterogeneous permeability fields extracted from the 10^{th} SPE comparative solution project (SPE10) [38], i.e., the top and the bottom layers. The SPE10 top is characterized by a smoothly varying permeability field (Fig. 4, left), whereas the SPE10 bottom has a channelized permeability field (Fig. 4, right) with long-correlation and high-contrast features, which are particularly challenging for multiscale solvers.

The saturation fields at 0.25 PVI (pore volume injected) obtained with the fine-scale solver are shown in Fig. 5a. We use these fields as references, s_{ref}, to evaluate the accuracy of the multiscale solutions. The saturation fields obtained with the standard MsFV method (without adaptivity) are shown in Fig. 5b; in these simulations, the multiscale global pressure system is updated and solved at every iteration of the sequentially implicit algorithm (in average about 3 iterations per time step). To highlight the saturation distribution errors, which are hardly visible in Fig. 5, we plot the difference between the two saturation fields (Fig. 5c). We can observe that errors are larger for the SPE10 bottom field (Fig. 5, right) than for the SPE10 top field (Fig. 5, left) due to the fact that the localization assumptions for interpolators are less accurate in presence of permeability structures that are characterized by high contrasts and long correlations.

To compare two time-dependent solutions we define the average L_2-error for the saturation, which is calculated as

$$\Delta S = \frac{1}{T}\int_0^T \|s(t) - s_{\text{ref}}(t)\|_2 \, dt, \tag{26}$$

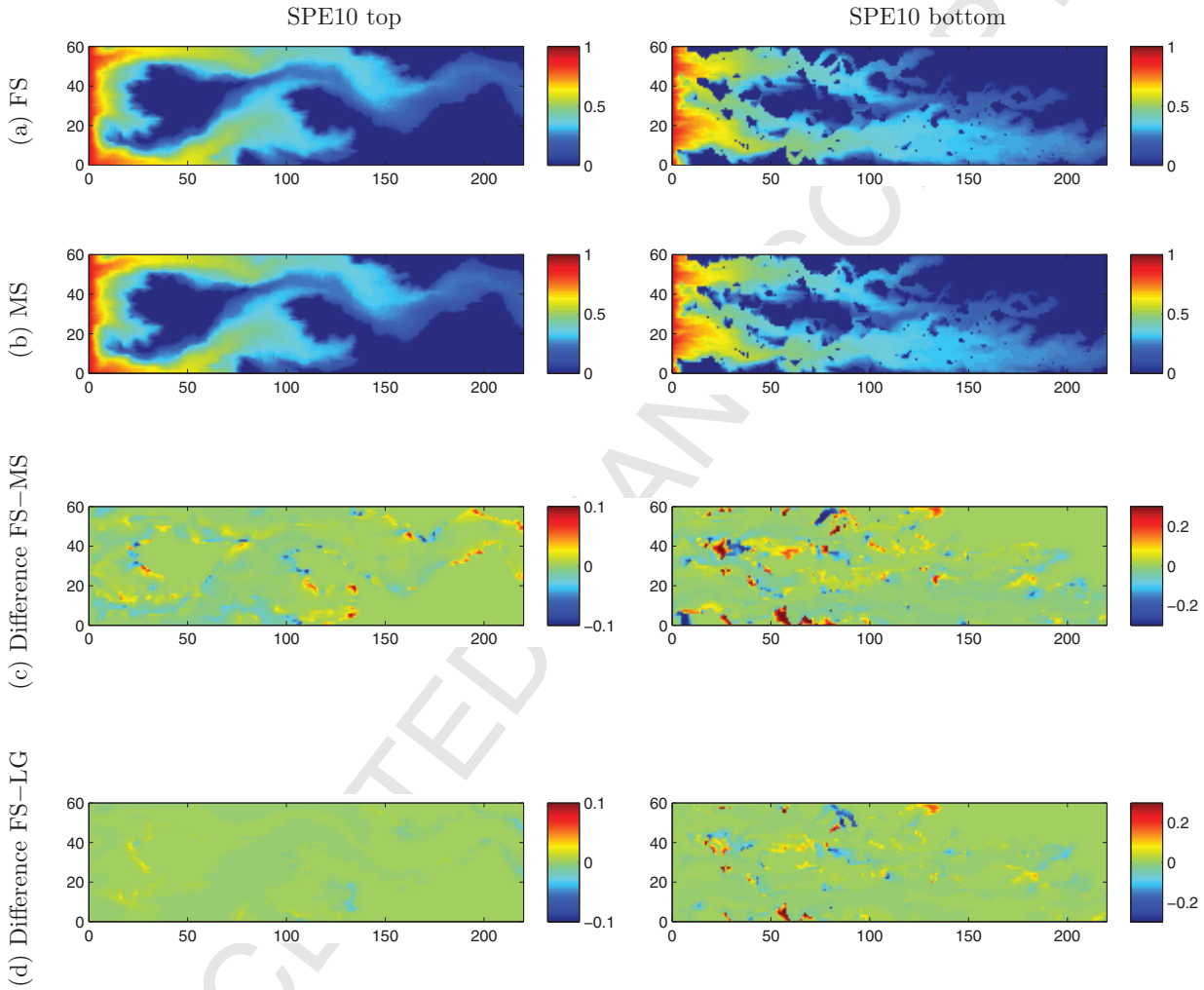


Figure 5: Saturation fields at PVI=0.25 for the SPE10 top (left) and SPE10 bottom (right) problems: (a) fine-scale reference solution (FS); (b) standard MsFV solution (MS); (c) saturation difference between the standard MsFV solution (b) and the reference (a); (d) saturation difference between spatiotemporal-adaptive algorithm with self-correction (LG) and the reference (a).

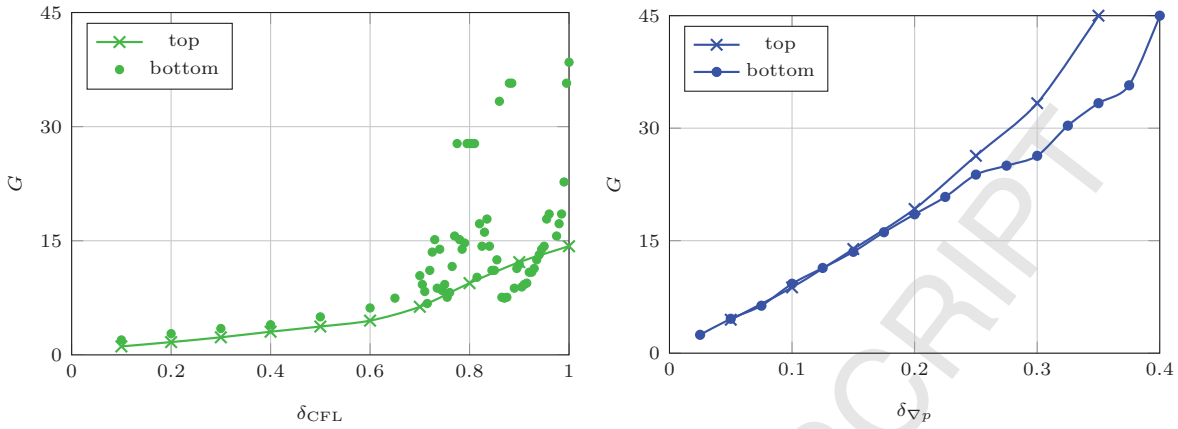


Figure 6: Global-local time step ratio, G, as a function of adaptive threshold for CFL (left) and gradient-based (right) criteria. For SPE10 bottom problem the performance of the CFL-criteria becomes unpredictable for $\delta_{\mathrm{CFL}} \gtrsim 0.7$.

where T is the total time.

First we study the efficiency and the accuracy of the new algorithm with respect to the temporal adaptivity parameters δ_{CFL} and $\delta_{\nabla p}$. The efficiency of the adaptive algorithm directly depends on the time step ratio, G. In Fig. 6, we plot G as a function of the adaptivity parameters δ_{CFL} and $\delta_{\nabla p}$. In general, relaxing δ_{CFL} or $\delta_{\nabla p}$ leads to larger values of G; however for SPE10 bottom problem, the CFL criterion does not guarantee a control on the efficiency for $\delta_{\mathrm{CFL}} \gtrsim 0.7$ (green dots in Fig. 6, left). This shows the advantages of the gradient-based criterion, for which smooth and monotonic relationship between G and $\delta_{\nabla p}$ is observed for both problems (Fig. 6, right).

To highlight the error stemming from adaptivity, we plot the difference between the average saturation error of the local-global algorithm and the average saturation error of the standard MsFV method, $\Delta S_{\mathrm{LG}} - \Delta S_{\mathrm{MS}}$ (Fig. 7). For the SPE10 bottom field the error generally increases with G, whereas for the SPE10 top field the error initially decreases showing a moderate optimum around $G = 3.7$ (resp. $G = 8.8$), which corresponds to $\delta_{\mathrm{CFL}} = 0.5$ (resp. $\delta_{\nabla p} = 0.1$). For both fields, the gradient-based criterion allows significantly larger values of G for the same accuracy (except few haphazard points in Fig. 7, right). This demonstrates that the gradient-based criterion is more robust than the CFL criterion, and it is superior in detecting whether coarse-scale mobility changes require updating the boundary condition of the local problems.

In Fig. 8 we plot the L^2-norm of the saturation error as a function of time (PVI). For SPE10 top problem (Fig. 8, left), we observe that for $\delta_{\mathrm{CFL}} = 0.5$ or $\delta_{\nabla p} = 0.1$ the accuracy of the the spatiotemporal-adaptive algorithm is roughly the same as the accuracy of the standard MsFV method, but the computational costs are lower due to the reduced number of global updates. With the CFL criterion ($\delta_{\mathrm{CFL}} = 0.5$), the global solution is updated in average every $G = 3.7$ local time steps without substantial accuracy loss. The

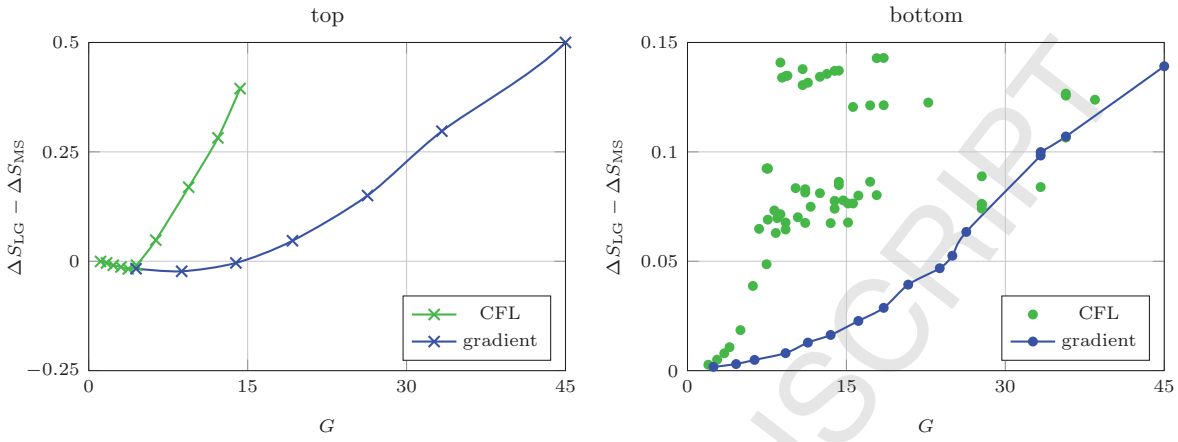


Figure 7: Difference between the average saturation errors, $\Delta S_{\mathrm{LG}} - \Delta S_{\mathrm{MS}}$, as a function of time step ratio, G, for SPE10 top (left) and bottom (right) problems. (Notice that negative values correspond to a more accurate adaptive solution, whereas for positive values the standard MsFV algorithm is more accurate.)

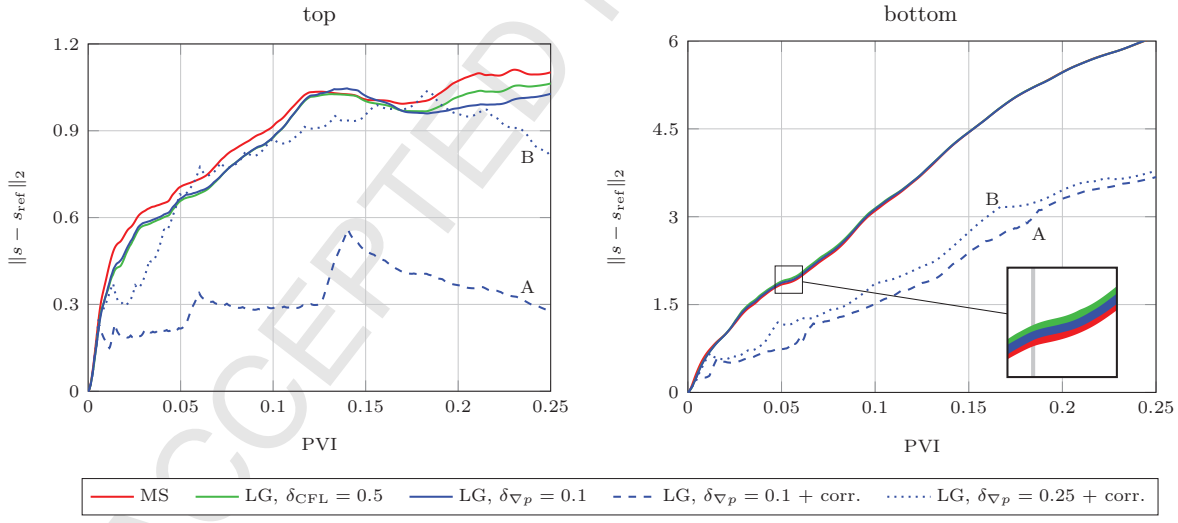


Figure 8: Evolution of the saturation error (expressed as L_2-norm of the saturation difference) with time (expressed in Pore Volume Injected, PVI) for SPE10 top (left) and SPE10 bottom (right) problems. The standard MsFV results (MS) are compared to the solution of the spatiotemporal-adaptive algorithm (LG) for different adaptive criteria. The error of simulations employing the self-correction (corr.) are also shown (A and B).

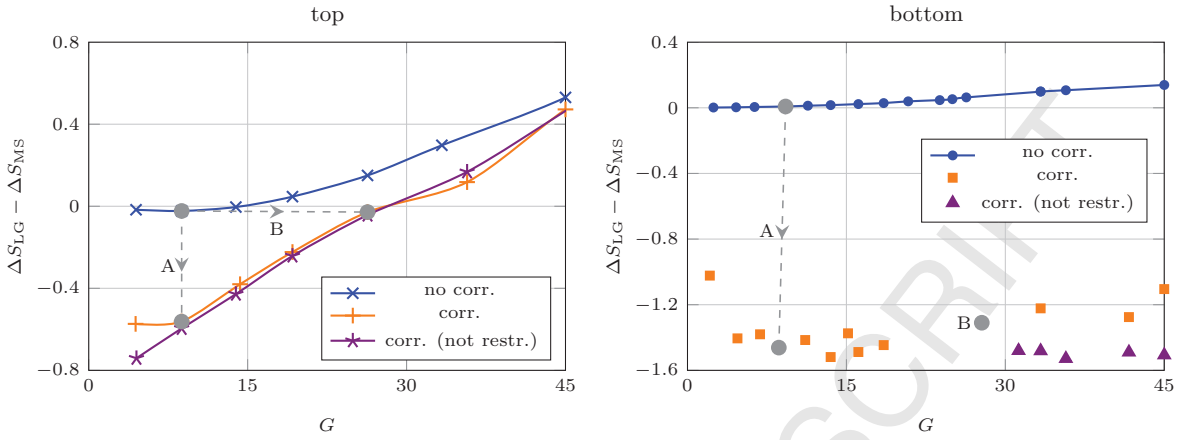


Figure 9: Difference between the average saturation errors, $\Delta S_{\mathrm{LG}} - \Delta S_{\mathrm{MS}}$, as a function of time step ratio, G, for SPE10 top (left) and bottom (right) problems. The gradient-based criterion is used. Blue points are obtained without self-correction (the same as in Fig.7), orange points are obtained with self-correction employed together with stability criterion Eq. 25, and violet points represents unrestricted self-correction (unstable for SPE10 bottom for $G \lesssim 30$). (Negative values correspond to a more accurate adaptive solution, whereas for positive values the standard MsFV algorithm is more accurate.)

gradient-based criterion is more effective and for $\delta_{\nabla p} = 0.1$ we have a solution roughly as accurate as the standard MsFV solution with one global update only every $G = 8.8$ local time steps. For the SPE10 bottom problem (Fig. 8, right), the criteria $\delta_{\mathrm{CFL}} = 0.5$ and $\delta_{\nabla p} = 0.1$ allow updating the global solution every $G = 5.0$ and $G = 9.2$ local time steps, respectively, with only a negligible deterioration of the solution.

Next we study the performance of the self-correcting algorithm together with the gradient-based criterion. For SPE10 top problem (Fig. 9, left), the self-correction yields a substantial accuracy improvement up to the $G \approx 30$. For very heterogeneous permeability fields, such as SPE10 bottom, the self-correcting scheme leads to a considerable improvement of the solution even for larger vales of G; this permits to substantially enhance the computational gain of the algorithm because the adaptive criterion can be relaxed (Fig. 9, right). This results from the improved localization assumptions used to define basis and correction functions. By improving the localization assumptions across dual cells, the self-correcting scheme allows reducing the multiscale approximation error, ϵ_{ms}. However, the total error of the adaptive algorithm (with respect to the fine-scale reference) results from the combined effects of the multiscale error and the error introduced by adaptivity, ϵ_{ad}. In presence of highly channelized flow, the medium heterogeneity dominates the flow (also in presence of multiple phases) and ϵ_{ms} is large. Although the self-correcting scheme requires several local time steps to ensure stability and may increase the adaptivity error, ϵ_{ad}, it effectively reduces the multiscale error, ϵ_{ms}, by using information about fluxes across dual cells, which increases the overall coupling between the global and the local scales. In case of channelized permeability, the heterogeneity structure of the medium (which is constant in time) dictates the coupling between local and global scales, which is not sensitively

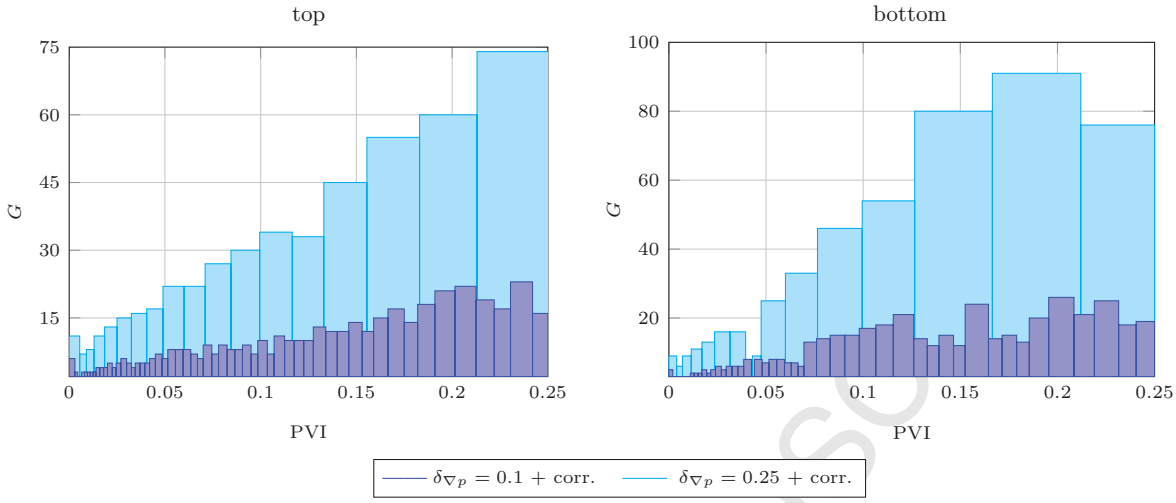


Figure 10: Evolution of time step ratio, G, with PVI for SPE10 top (left) and bottom (right) problems. Scenarios A and B from Fig. 9 are demonstrated. The width of the bars indicates the coarse time-step interval in PVI, and the height indicates the number of fine-scale time steps in the coarse interval.

affected by the phase distribution (which varies with time). Therefore, the reduction of the multiscale error dominates the increase of the adaptive error related to a less frequent update of the boundary conditions.

Although for relatively simple cases (e.g., SPE10 top problem) the self-correcting scheme is stable for all time step ratios, G, it may diverge for more challenging problems and the stability criterion, Eq. 25, has to be employed. In the SPE10 bottom problem, for example, the scheme is stable only for $G \gtrsim 30$. The criterion in Eq. 25 provides a sufficient condition to stabilize the scheme for $G < 30$.

When the self-correcting scheme is used together with the adaptivity threshold $\delta_{\nabla p} = 0.1$ for SPE10 top problem, the final saturation error decreases by more than 60% (scenario A in Fig. 9, left; see also Fig. 5d and Fig. 8) while the ratio between time steps remains the same $G = 8.8$. On the other hand, if the error of the standard MsFV algorithm is acceptable (scenario B in Fig. 9, left), the self-correcting scheme allows us to improve the efficiency by relaxing the adaptive criterion without accuracy loss: if $\delta_{\nabla p} = 0.25$ is used together with the self-correction, approximately the same accuracy of the standard MsFV method is obtained updating the global problem only every $G = 26.3$ local time steps (Fig. 8, left).

For the more challenging SPE10 bottom problem, the self-correcting scheme together with the gradient-based criterion $\delta_{\nabla p} = 0.1$ gives a reduction of the saturation error around 45% (scenario A in Fig. 9, right; see also Fig. 5d and Fig. 8), while the ratio between global and local time steps only slightly decreases ($G = 8.6$). The self-correcting scheme substantially improves the accuracy of the solution, allowing very relaxed adaptivity criteria (e.g., scenario B in Fig. 9, right; see also Fig. 8). Notice that in all simulations the global-local time step ratio, G, is not constant but tends to increase with time (Fig. 10).

Table 1: Pore geometry properties.

Domain size	88 mm × 114 mm
Mean obstacle size	0.6 mm
Number of obstacles	10 028
Number of cells	2 767 077
Permeability	10^{-8} m^2
Porosity	68%

4.2. Multiphysics description of two-phase flow

We simulate the drainage of two fluid pairs with two different viscosity ratios: a favorable, $M = \mu_{\mathrm{nw}}/\mu_{\mathrm{w}} = 3.0$, and an unfavorable ratio, $M = 0.2$. We consider an idealized porous medium consisting of a rectangular two-dimensional horizontal domain which is filled with impermeable circular obstacles (see Tab. 1 and [26] for the details). The non-wetting phase is injected at constant velocity, $u_{\mathrm{in}} = 10^{-2}$ m/s, whereas a constant pressure, $p_{\mathrm{out}} = 0$, is assigned at the outlet. The pore-scale Reynolds and capillary numbers resulting from the simulation parameters are Re $\sim 10^{-2}$ and Ca $\sim 10^{-1}$, respectively. The pore-scale simulations are performed with a modified implementation of the open-source software package OpenFOAM [39].

The domain is discretized into a fine grid consisting of about 3 millions cells (see Tab. 1), whereas the auxiliary coarse grid has 7×8 nodes, which yields an upscaling factor $\Upsilon \approx 5 \cdot 10^4$ and to a ratio between coarse cell size and fine cell size $\epsilon^{-1} = H_c/h_f \approx \Upsilon^{1/d} \approx 2 \cdot 10^2$. The adaptive multiscale solution is compared with the reference pore-scale solution obtained by solving the same problem directly on the fine grid. Notice that, in addition to the localization assumption to compute the local interpolators, it is assumed that the coarse-scale fluxes depend only on the global pressure (Eq. 8); this hypothesis is justified in case of sufficiently large upscaling factors (see [16] for a detailed discussion).

First, we demonstrate the effects of adaptivity on the accuracy of the solution. We employ the gradient-based criteria which has proved effective and robust for Darcy-scale problems.

As it is expected that for unstable flow regimes errors introduced by the localization might grow leading to different flow distribution, we evaluate the quality of the multiscale solutions in a statistical sense rather than in a deterministic (pointwise) sense (see also [8]). The transversally averaged saturation profiles are plotted in Fig. 11 for different values of the time-step ratio G, which result from different adaptive parameters $\delta_{\nabla p}$. We observe that varying G mainly affects the penetration depth. For the stable flow regime (Fig. 11, left) the penetration depth increases with G and the average profile becomes more dispersed: adaptivity slightly perturbs the front at the scale of coarse grid. For the unstable regime (Fig. 11, right) the situation is reversed: the front advances more slowly for larger G. Due to viscosity difference, the fluid velocity is faster in the fingers, but more frequent global updates are required to fully capture this effects at the global

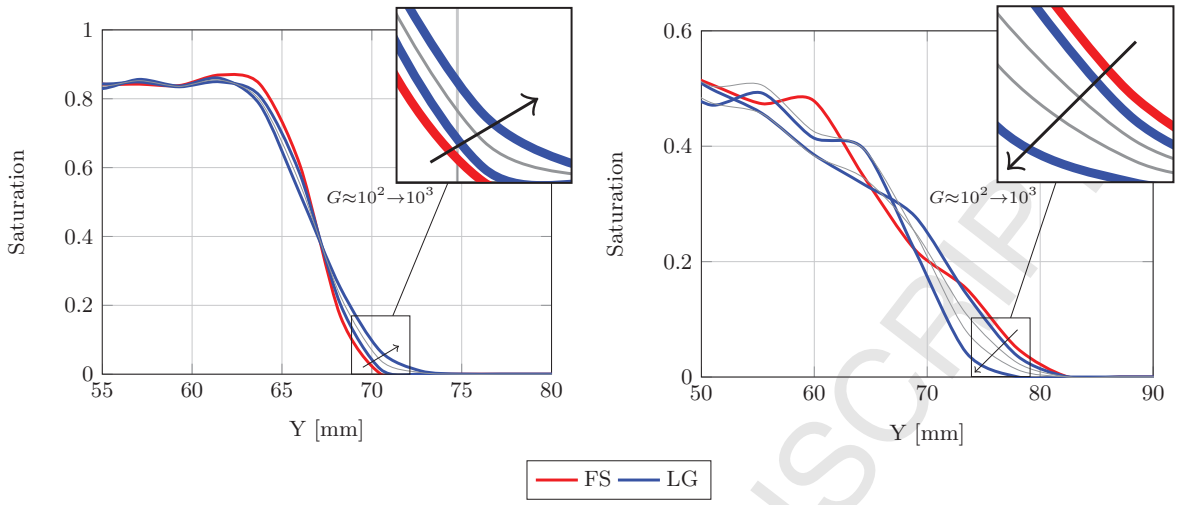


Figure 11: Dependency of the accuracy of the adaptive algorithm on global-local time step ratio G for the stable ($M = 3$, left) and the unstable ($M = 0.2$, right) flow regimes. The average saturation profile in the front region is shown. The pore-scale reference solution (FS) is shown in red, whereas the blue lines correspond to the results of the adaptive simulations (LG) for the two extreme values of G employed (i.e., $G = 10^2$ and $G = 10^3$) whereas the gray lines correspond to intermediate values of G. The arrows indicate the direction of growth of the time step ratio G.

scale; relaxing the adaptivity parameters reduces the degree of coupling at the global scale.

The results of the spatiotemporal-adaptive algorithm are compared with the reference solution in Fig. 12, which shows the phase distributions and the average saturation profiles for the two viscosity ratios. We show the results obtained with $\delta_{\nabla p} = 0.1$ for $M = 3.0$ and $\delta_{\nabla p} = 0.2$ for $M = 0.2$, which both correspond to $G \approx 10^2$. For favorable viscosity ratio ($M = 3$) the flow is stable and the multiscale solution is in excellent agreement with the reference. In the unfavorable case ($M = 0.2$) the flow becomes unstable and elongated structures emerge in the fluid distribution. The adaptive algorithm captures the transition to the unstable regime and the average saturation profile is very well reproduced. Notice that an exact (deterministic) reproduction of the fluid pattern is impossible in the unstable regime because small errors in the approximate velocity field are amplified with time; therefore, only a statistic reproduction of the flow patterns is expected (see, e.g., [8]).

The comparison of the saturation profiles shows that the adaptive algorithm captures the main characteristics of the flow (e.g., penetration depth and variability). The results of the spatiotemporal-adaptive algorithm have been obtained with a ratio between global and local time steps of about $G \approx 10^2$, which can substantially improve the efficiency of the multiscale approach in case of problems that involve a very large number of pores as it is expected in realistic three-dimensional applications.

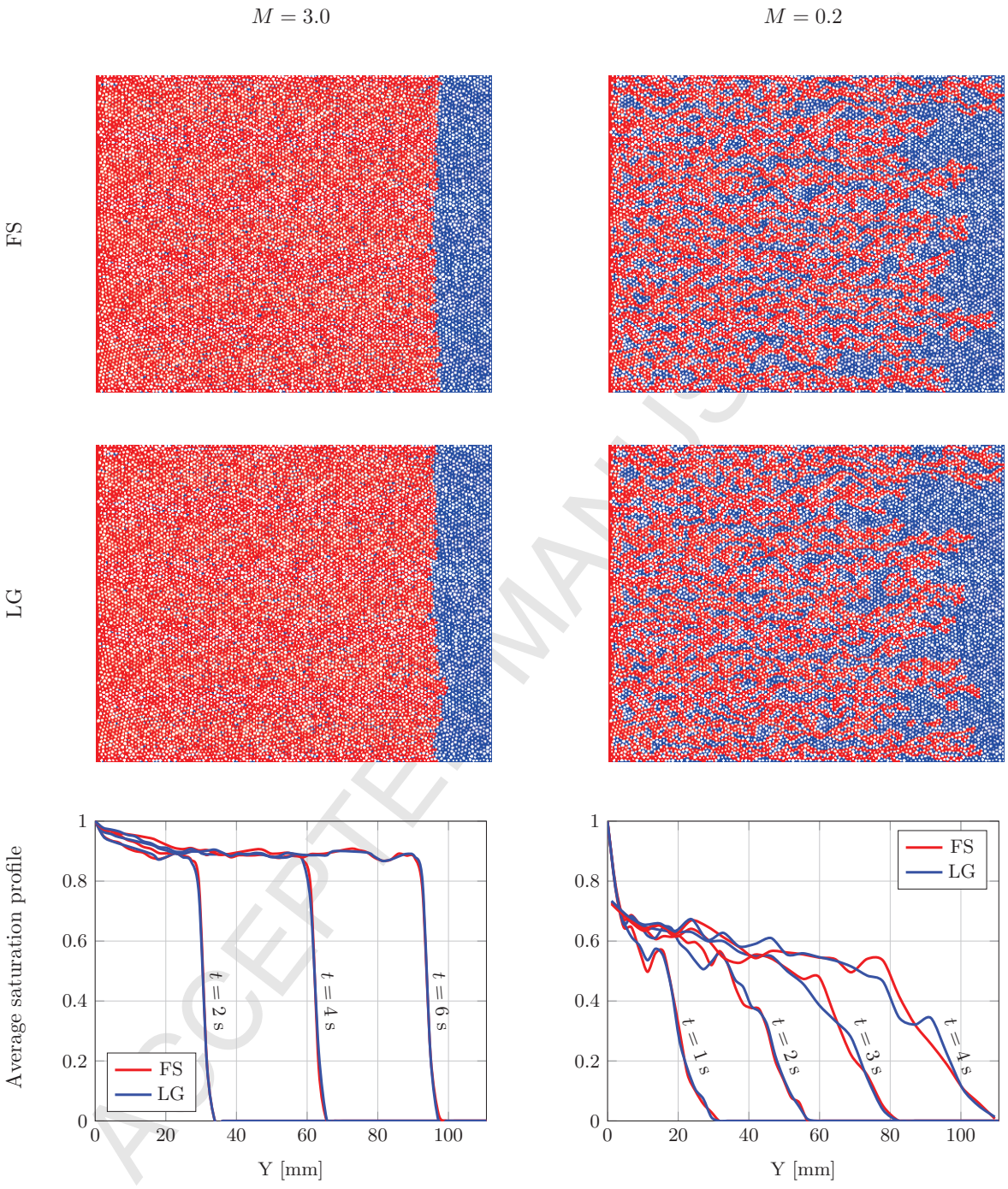


Figure 12: Comparison of the pore-scale reference solutions (FS) and the multiphysics adaptive solutions (LG). We have used $\delta_{\nabla p} = 0.1$ for the stable flow regime ($M = 3.0$, left); and $\delta_{\nabla p} = 0.2$ for the unstable flow regime ($M = 0.2$, right). For both cases the adaptive parameter yields a time-step ratio $G \approx 10^2$.

5. Conclusions

The standard MsFV formulation for time-dependent problems (typically Darcy-scale flow and transport in porous media) begins by splitting the problem into a pressure part and a transport part. The MsFV method is then applied to the pressure equation, whereas the transport equation is solved separately. In this context the MsFV method is regarded as an efficient algorithm to approximate the solution of an elliptic problem. We have proposed a different interpretation that relies on a local-global splitting in which the full system of equations is solved locally. This formulation allows retaining the original level of coupling between the different equations and offers several advantages.

This splitting provides great flexibility in terms of spatiotemporal adaptivity and naturally allows using different time steps for the global and the local problems as a direct consequence of the different spatial scales. To decide when the global problem has to be solved to update the boundary conditions of the local problems, we employ adaptive criteria based on global quantities rather than on fine-scale quantities as routinely done the MsFV method. We have devised a criterion based on monitoring the relative changes of the pressure force on coarse-cell boundaries to provide an estimate of coarse-scale mobility changes. This criterion allows employing considerably larger time steps for the global problem than for the local problems without a substantial loss of accuracy.

A simple complexity analysis demonstrates a noticeable speed-up with respect to the standard MsFV algorithm. In particular, the spatiotemporal adaptivity allows reducing the computational costs and keeping the multiscale method efficient with respect to iterative solvers in case of large upscaling factors. This is particularly important in case of local problems involving a large number of degrees of freedom as they can be encountered in multiphysics applications, which cannot be handled efficiently by the standard MsFV method (see also [31]).

The use of different time steps for the local and global problems does not only permit to improve efficiency, but also accuracy. We have devised a self-correcting algorithm that improves the quality of the solution by using information from the previous time to estimate the transversal fluxes across dual-cell boundaries, which are neglected to compute basis and correction functions in the MsFV method. As the local time steps act as smoothing steps, this correction scheme is stable only if the number of local time steps per global time step is sufficiently large. To prevent the correction scheme from diverging, the flux correction is applied only if its norm grows slower than exponentially with time. This leads to a win-win situation: simulation results demonstrate that the spatiotemporal adaptivity with self-correction yields more accurate multiscale solution at reduced computational costs.

The local-global splitting presented here gives new insight into the MsFV framework and enlarges the domain of applicability of the method, which can be employed to solve an arbitrary system of conservation equations. As the local problems solve the full system of equations with an original level of cou-

pling, the method is particularly suited for difficult problems such as unstable flow [8]. Also, the proposed spatiotemporal-adaptive framework is especially effective for multiphysics problems in which relatively large scale separation, and possibly the different nature of equations, might lead to different characteristic times. The framework has great potential for modeling multiphase flow in petroleum reservoirs [40] or during CO_2 storage [41], for applications to geomechanical problems [42, 43], for simulations of in-situ combustion process [44] or reactive transport [45, 46, 47]. In all these examples the temporal scales exhibit a wide range of variation, but standard splitting methods might prove inadequate because retaining a tight degree of local coupling is crucial. A spatiotemporal-adaptive algorithm is also of great value for parallel computing because it allows us to reduce the frequency at which information is exchanged among processors by reducing the frequency at which boundary conditions are updated. Finally, it might improve the performance of non-linear solvers since the size of the local sub-problems is reduced (see, e.g., [48]).

6. Acknowledgments

This project is supported by the Swiss National Science Foundation grants FNS PP00P2_123419 and PP00P2_144922. The authors wish to thank Rouven Künze for many constructive discussions. Also, we thank the anonymous reviewers whose constructive comments helped to improve the clarity of the manuscript.

Appendix A. Governing equations

We consider the flow of two immiscible incompressible fluids through a non-deformable porous medium. For simplicity (but without loss of generality) we deal with horizontal two-dimensional domains, and we neglect gravity.

Appendix A.1. Two-phase flow at the Darcy scale

The mass-balance equations for the two phases read

$$\phi \frac{\partial s_\beta}{\partial t} + \nabla \cdot \boldsymbol{u}_\beta = q_\beta, \quad \beta = 1, 2, \tag{A.1}$$

where ϕ is the porosity, s_β the phase saturation, q_β the source or sink term, and

$$\boldsymbol{u}_\beta = -\lambda_\beta \nabla p \tag{A.2}$$

the Darcy velocity of the phase β. In Eq. A.2, p is the pressure (we neglect capillary effects), $\lambda_\beta = k_{\mathrm{r}\beta}\boldsymbol{k}/\mu_\beta$ the phase mobility, $k_{\mathrm{r}\beta}$ the relative permeability, $\boldsymbol{k}$ the absolute permeability tensor, and μ_β the viscosity of the phase.

Given the dependency of the relative permeability on the saturation, $k_{r\beta} = k_{r\beta}(s_\beta)$, and the volume constraint $s_1 + s_2 = 1$, we have a complete system of equations for the unknown variables, p, s_1, and s_2.

We employ the pressure-saturation formulation. By summing up the two mass-balance equations, Eq. A.1, we obtain the pressure equation

$$-\nabla \cdot (\lambda_t \nabla p) = q_t, \tag{A.3}$$

where $\lambda_t = \lambda_1 + \lambda_2$ is the total mobility, and $q_t = q_1 + q_2$ the total source term. The fractional flow formulation is used for the saturation equation

$$\phi \frac{\partial s}{\partial t} + \nabla \cdot (F(s)\boldsymbol{u}_t) = q, \tag{A.4}$$

where $s = s_1$, $F(s) = \lambda_1/\lambda_t$, $\boldsymbol{u}_t = \boldsymbol{u}_1 + \boldsymbol{u}_2$, $q = q_1$.

Appendix A.2. Pore-scale description of two-phase flow

At the pore-scale, we use the whole-domain formulation (see, e.g., [24]), in which the two phases are represented as a single fluid with variable properties. The fluid distribution is described by the phase indicator α, which is the volumetric fraction of the phase 1 and varies between 0 and 1.

The density, ρ, and the viscosity, μ, are functions of α (which is space dependent), i.e.,

$$\rho = \alpha\rho_1 + (1-\alpha)\rho_2, \qquad \mu = \alpha\mu_1 + (1-\alpha)\mu_2, \tag{A.5}$$

where ρ_1, ρ_2, μ_1, and μ_2 are the constant densities and viscosities of the fluids.

The surface force (per unit volume) that acts at the fluid-fluid interface is

$$\boldsymbol{f}_s \approx \gamma\kappa\nabla\alpha, \tag{A.6}$$

where γ is the surface tension, and $\kappa = -\nabla \cdot \boldsymbol{n} = \nabla \cdot (\nabla\alpha/|\nabla\alpha|)$ the total curvature of the interface.

Finally, for pressure, p, velocity, $\boldsymbol{u}$, and color function, α, we have the following system of equations

$$\nabla \cdot \boldsymbol{u} = 0, \tag{A.7}$$

$$\frac{\partial \rho\boldsymbol{u}}{\partial t} = -\nabla p + \nabla \cdot \mu\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T\right) + \gamma\kappa\nabla\alpha, \tag{A.8}$$

$$\frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha\boldsymbol{u}) = 0. \tag{A.9}$$

(The non-linear term in momentum equation is neglected since we consider the flow at low Reynolds number, see, e.g., [16] and references therein.)

At the solid boundary, $\partial\Omega^s$, the standard no-slip boundary condition is imposed for $\boldsymbol{u}$ and the normal to the fluid-fluid interface is assigned as boundary condition to account for contact-angle effects. The pore-scale simulations are performed with a modified implementation of the OpenFOAM [39].

References

[1] J. A. Daws, D. J. Prosser, Scales of permeability heterogeneity within the Brent Group, Journal of Petroleum Geology 15 (3) (July 1992) 397–418.

[2] Y. Guieguen, R. Gavrilenko, M. Le Ravalec, Scales of rock permeability, Surveys in Geophysics 17 (1996) 245–263.

[3] P. Jenny, S. H. Lee, H. A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, Journal of Computational Physics 187 (1) (2003) 47–67.

[4] I. Lunati, P. Jenny, Multi-scale finite-volume method for compressible flow in porous media, Journal of Computational Physics 216 (2) (2006) 616–636.

[5] H. Hajibeygi, P. Jenny, Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media, Journal of Computational Physics 228 (14) (2009) 5129–5147.

[6] I. Lunati, P. Jenny, Multiscale finite-volume method for density-driven flow in porous media, Computational Geosciences 12 (3) (2008) 337–350.

[7] R. Künze, I. Lunati, An adaptive multiscale method for density-driven instabilities, Journal of Computational Physics 231 (2012) 5557–5570.

[8] R. Künze, P. Tomin, I. Lunati, Local modeling of instability onset for global finger evolution, Advances in Water Resources 70 (2014) 148–159.

[9] S. H. Lee, C. Wolfsteiner, H. A. Tchelepi, Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity, Computational Geosciences 12 (3) (2008) 351–366.

[10] H. Hajibeygi, H. A. Tchelepi, Compositional multiscale finite-volume formulation, SPE-163664-PA, SPE Journal 19 (2) (2014) 316–326.

[11] C. Wolfsteiner, S. H. Lee, H. A. Tchelepi, Well modeling in the multiscale finite volume method for subsurface flow simulation, Multiscale Modeling and Simulation 5 (3) (2006) 900–917.

[12] P. Jenny, I. Lunati, Modeling complex wells with the multi-scale finite-volume method, Journal of Computational Physics 228 (3) (2009) 687–702.

[13] H. Hajibeygi, D. Karvounis, P. Jenny, A hierarchical fracture model for the iterative multiscale finite volume method, Journal of Computational Physics 230 (24) (2011) 8729–8743.

[14] T. H. Sandve, E. Keilegavlen, J. M. Nordbotten, Physics-based preconditioners for flow in fractured porous media, Water Resources Research 50 (2) (February 2014) 1357–1373.

[15] G. Bonfigli, P. Jenny, An efficient multi-scale Poisson solver for the incompressible Navier-Stokes equations with immersed boundaries, Journal of Computational Physics 228 (12) (2009) 4568–4587.

[16] P. Tomin, I. Lunati, Hybrid multiscale finite volume method for two-phase flow in porous media, Journal of Computational Physics 250 (1 October 2013) 293–307.

[17] P. Jenny, S. H. Lee, H. A. Tchelepi, Adaptive multiscale finite-volume method for multi-phase flow and transport in porous media, Multiscale Modeling and Simulation 3 (1) (2004) 50–64.

[18] P. Jenny, S. H. Lee, H. Tchelepi, Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media, Journal of Computational Physics 217 (2) (2006) 627–641.

[19] H. Hajibeygi, P. Jenny, Adaptive iterative multiscale finite volume method, Journal of Computational Physics 230 (3) (2011) 628–643.

[20] S. H. Lee, H. Zhou, H. A. Tchelepi, Adaptive multiscale finite-volume method for nonlinear multiphase transport in heterogenous formations, Journal of Computational Physics 228 (24) (2009) 9036–9058.

[21] I. Lunati, S. Lee, An operator formulation of the multiscale finite-volume method with correction function, Multiscale Modeling and Simulation 8 (1) (2009) 96–109.

[22] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1) (1981) 201–225.

[23] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, Journal of Computational Physics 113 (1994) 134–147.

[24] R. Scardovelli, S. Zalesky, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1999) 567–603.

[25] I. Lunati, D. Or, Gravity-driven slug motion in capillary tubes, Physics of Fluids 21 (5) (2009) 052003.

[26] A. Ferrari, I. Lunati, Direct numerical simulations of interface dynamics to link capillary pressure and total surface energy, Advances in Water Resources 57 (July 2013) 19–31.

[27] A. Q. Raeini, M. J. Blunt, B. Bijeljic, Direct simulations of two-phase flow on micro-CT images of porous media and upscaling of pore-scale forces, Advances in Water Resources.

[28] R. Helmig, J. Niessner, B. Flemisch, M. Wolff, J. Fritz, Efficient modelling of flow and transport in porous media using multi-physics and multi-scale approaches, Preprint Series Issue No. 2010-27, Stuttgart Research Centre for Simulation Technology (SRC SimTech).

[29] A. Tartakovsky, T. Scheibe, Dimension reduction numerical closure method for advection-diffusion-reaction systems, Advances in Water Resources 34 (2011) 1616–1626.

[30] V. Kippe, J. E. Aarnes, K.-A. Lie, A comparison of multiscale methods for elliptic problems in porous media flow, Computational Geosciences 12 (2008) 377–398.

[31] R. Künze, I. Lunati, S. H. Lee, The multilevel multiscale finite volume method, Journal of Computational Physics 255 (2013) 502–520.

[32] H. Hajibeygi, G. Bonfigli, M. A. Hesse, P. Jenny, Iterative multiscale finite-volume method, Journal of Computational Physics 227 (19) (2008) 8604–8621.

[33] I. Lunati, M. Tyagi, S. H. Lee, An iterative multiscale finite volume algorithm converging to the exact solution, Journal of Computational Physics 230 (2011) 1849–1864.

[34] H. Hajibeygi, S. Lee, I. Lunati, Accurate and efficient simulation of multiphase flow in a heterogeneous reservoir by using error estimate and control in the multiscale finite-volume framework, SPE Journal, SPE-141954-PA 17 (4) (2012) 107—-1083.

[35] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856–869.

[36] Y. Saad, Iterative methods for sparse linear systems, Siam, 2003.

[37] K. H. Coats, A note on IMPES and some IMPES-based models, Society of Petroleum Engineers Journal 5 (2000) 245–251.

[38] M. A. Christie, M. J. Blunt, Tenth SPE comparative solution project: A comparison of upscaling techniques, SPE Reservoir Evaluation & Engineering 4 (4) (2001) 308–317.

[39] OpenFOAM, The Open Source CFD Toolbox: User Guide, OpenFOAM Foundation, version 2.1.1 (2012).

[40] A. K. Pergament, V. A. Semiletov, P. Y. Tomin, On some multiscale algorithms for sector modeling in multiphase flow in porous media, Mathematical Models and Computer Simulations 3 (3) (2011) 365–374.

[41] B. Guo, K. W. Bandilla, F. Doster, E. Keilegavlen, M. A. Celia, A vertically integrated model with vertical dynamics for CO_2 storage, Water Resources Research.

[42] J. Kim, H. A. Tchelepi, R. Juanes, Rigorous coupling of geomechanics and multiphase flow with strong capillarity, SPE Journal 18 (6) (2013) 1–123.

[43] M. A. Murad, M. Borges, J. A. Obregón, M. Correa, A new locally conservative numerical method for two-phase flow in heterogeneous poroelastic media, Computers and Geotechnics 48 (2013) 192–207.

[44] R. Younis, M. Gerritsen, et al., Multiscale process coupling by adaptive fractional-stepping; an in-situ combustion model,

SPE-93458-MS, in: SPE/DOE Symposium on Improved Oil Recovery, Society of Petroleum Engineers, 2006.

[45] I. Battiato, D. M. Tartakovsky, A. M. Tartakovsky, T. D. Scheibe, Hybrid models of reactive transport in porous and fractured media, Advances in Water Resources 34 (9) (2011) 1140–1150.

[46] D. Roubinet, D. M. Tartakovsky, Hybrid modeling of heterogeneous geochemical reactions in fractured porous media, Water Resources Research 49 (12) (2013) 7945–7956.

[47] Y. Mehmani, T. Sun, M. Balhoff, P. Eichhubl, S. Bryant, Multiblock pore-scale modeling and upscaling of reactive transport: application to carbon sequestration, Transport in porous media 95 (2) (2012) 305–326.

[48] J. O. Skogestad, E. Keilegavlen, J. M. Nordbotten, Domain decomposition strategies for nonlinear flow problems in porous media, Journal of Computational Physics 234 (1 February 2013) 439–451.