Politecnico di Milano

POLITECNICO
MILANO 1863

Department of Mathematics
Mathematical models and methods in engineering
MOX - Laboratory for Modeling and Scientific Computing

PhD thesis

# Conservative multirate schemes for flow in heterogeneous porous media

Ludovica Delpopolo Carciopolo

*Supervisor* Prof. Luca Formaggia

*Co-Advisors* Dr. Anna Scotti

Prof. Hadi Hajibeygi

**Ludovica Delpopolo Carciopolo**
*Conservative multirate schemes for flow in heterogeneous porous media*
PhD thesis, October 30, 2019
Supervisor: Prof. Luca Formaggia
Co-Advisors: Dr. Anna Scotti, Prof. Hadi Hajibeygi

**Politecnico di Milano**
*MOX - Laboratory for Modeling and Scientific Computing*
Department of Mathematics
Mathematical models and methods in engineering
p.zza Leonardo da Vinci, 32
20133 Milan

# Abstract

The aim of this work is to develop multirate time integration methods for multiphase flow in heterogeneous porous media. Numerical simulations of multiphase systems may involve many challenges: physical phenomena that take place at different space scales (from micro-scales to real field scales) and highly heterogeneous permeability fields that may cause the co-existence of fast and slow processes. For these reasons we propose a solution strategy based on a self-adjusting conservative implicit multirate method. The scheme is able to use different time step sizes in different area of the domain preserving the local mass conservation thanks to a flux-based refinement strategy. The properties of the proposed method, consistency and stability, are analyzed, showing that the multirate approach preserves the properties of the underlying numerical scheme used to integrate the system in time.

The novel multirate approach is combined with an advanced multi-scale technique. The obtained method is able to address the multiscale in space nature of the problem, and also to tackle multiple time scales for the sequentially coupled flow and transport equations.

Moreover, a space-time algebraic dynamic multilevel method (ADM-LTS) to solve transport equations in heterogeneous porous media is presented. The method constructs an adaptive multilevel space-time grid, based on error estimators, and solves the time dependent problem with local time stepping. The method employs space-time fine-grid cells only at the moving saturation fronts, reducing the computational costs and preserving the accuracy of the solution.

The accuracy and efficiency of the new methods are investigated in a wide range of numerical test cases for multiphase flows in heterogeneous porous media. The results demonstrate that the proposed methods provide a promising strategy to optimize the trade-off between accuracy and efficiency.

# Contents

# List of Figures

# List of Tables

# Introduction

# 1

Computational models of multiphase flows in porous media are of great importance in many areas of engineering – e.g., hydrology and groundwater flows, oil and gas reservoirs, geothermal energy, and waste management, including $CO_2$ sequestration in the subsurface.

In hydrology and groundwater applications, the quality of the water is often endangered by a large number of contaminants (e.g., atmospheric contaminants, leakage from storage tanks that may contain gasoline, oil or chemicals), as shown in Fig. 1.1. Multiphase flow models allow us to predict the flow of these contaminants, which may form a separate phase immiscible with water, by means of suitable equations [59].

Regarding oil reservoirs exploitation, an important technique for Hydrocarbon extractions is EOR – Enhanced Oil Recovery. EOR methods are based on the injection of different substances (thermal, gas, or chemicals) in the reservoir in order to increase the recovery factor of the oil. It is crucial to develop high-fidelity simulations of EOR in order to operate efficiently these advanced techniques.

One of the most relevant application of multiphase flow is geothermal energy production, which is strictly linked with renewable energy. The thermal energy is contained in the rocks and fluids from the shallow ground to several miles below the surface. Indeed, these underground reservoirs can be topped to generate electricity or used to directly heat and cool buildings. The energy can be extracted directly from the groundflow, with no combustions and nearly zero air emissions – see [88]. This complex scenario can be modeled by the equations of multiphase flow to understand the phenomena and increase the amount of green energy production.

Finally, $CO_2$ sequestration is a process used to capture and store the atmospheric carbon dioxide ($CO_2$). This carbon dioxide is the product of industrial processes and it can be stored in geological formations, such as subsurface saline aquifers, reservoirs, or coal seams – see Fig. 1.3. Moreover, the captured $CO_2$ can be also used for EOR to displace oil. Numerical methods are very important to study the physical mechanisms related to $CO_2$ sequestration, in particular to assess the safety of long-term storage by evaluating the risk of leakage.

All these geoengineering applications involve a number of physical phenomena that take place at different scales, from the micro-scale scale, where localized alteration of the rock properties may occur, to the large reservoir scale where the fluids displace. In fact, on one hand geological formations extend for several hundreds of meters; on the other hand, physical and chemical phenomena, which are of interest for the described applications, occur at much smaller scales (cm and below). Moreover, fast processes (e.g., high velocity flow in highly permeable rocks and fractures) and slow processes (e.g., flow in low permeable porous rocks) coexist, and they both have to be correctly represented to obtain reliable numerical simulations. Additionally, at the continuum (or Darcy) scale, porous media present highly

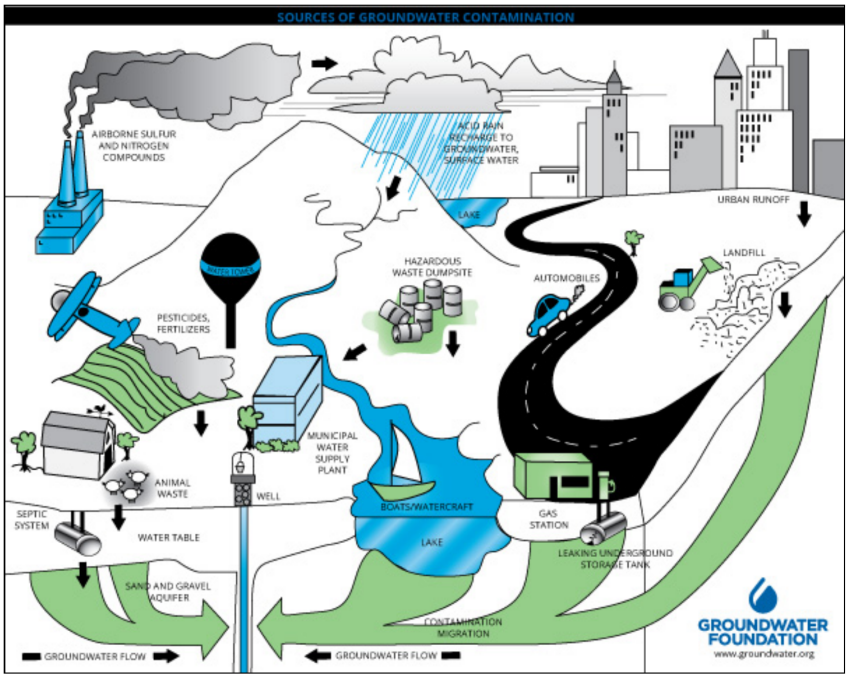


**Fig. 1.1.:** Source of groundwater contaminants (Figure from groundwater foundation website).

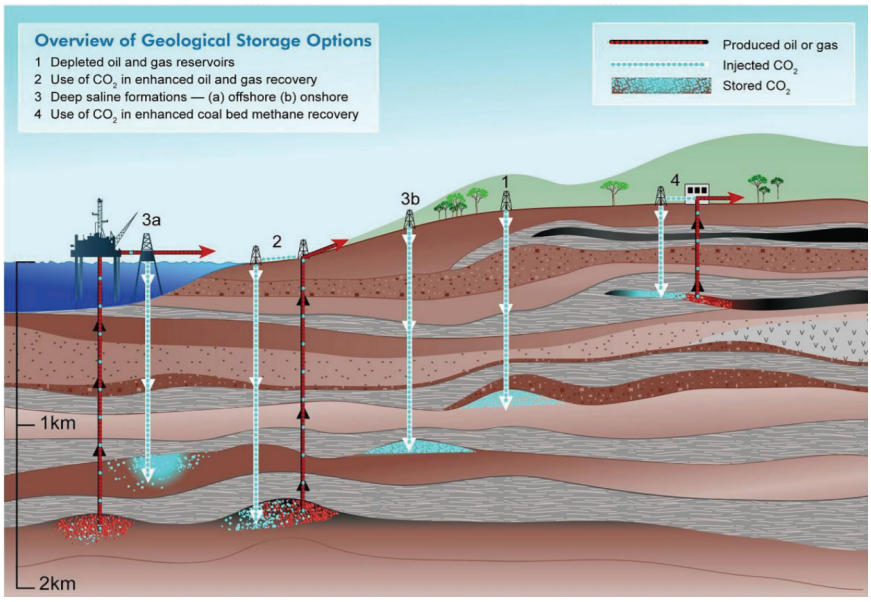


**Fig. 1.2.:** Gelogical storage options (Figure from [70]).

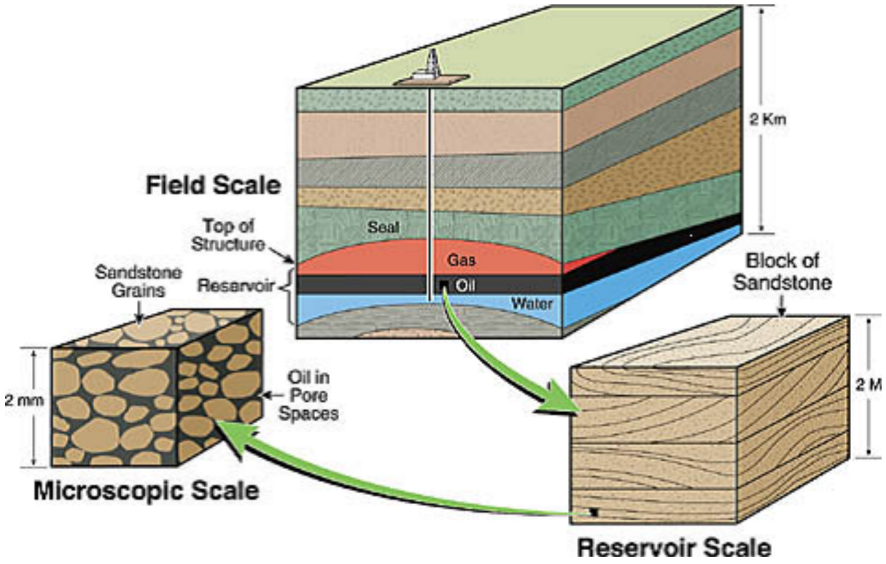


**Fig. 1.3.:** Different scales involved in geoengineering simulations (Figure from [102]).

heterogeneous permeability tensors which are a measure of the ability of the rock to conduct flow. Thus, accurate numerical models require very high resolution grids both in space and time to capture all the relevant physics of the phenomena. However, the large size of the domains and the large number of realizations needed to reduce uncertainty [13, 74], make field scale simulations impracticable on such high resolution grids.

Traditionally, upscaling techniques [27, 31, 30, 34] were used to reduce the computational costs. These methods map the rock and the fluid properties defined on a fine-scale grid to a coarser resolution. However, in presence of complex physics and in big ratio between the fine and the coarse resolution, the upscaled solution may be non-satisfactory. For this reason some advanced schemes that are able to employ higher resolution grids have been developed. An example are the dynamic local grid refinements (DLRG) methods [7, 36] that allow to use an adaptive grid resolution throughout the time dependent simulation. They need some pre-calculated upscaled static quantities and at each time step, when the grid resolution is determined, the linear system needs to be constructed. The well known multiscale methods [52, 33, 47] are another example of advanced algorithms that combines different scales and speed-up simulations. Finally, the algebraic dynamic multilevel method (ADM) [23, 24] is able to use a dynamic adaptive grid in space. Thanks to the restriction and prolongation operators, it does not need to construct the discrete linear system at each time step but it is able to use an automatic (algebraic) procedure which constructs the linear system directly from the fine resolution. However, all these approaches use a fixed time resolution grid at each time step and are not combined with any advanced time integration scheme.

On the other hand, the challenge of time adaptivity has been addressed by Adaptive Implicit methods (AIM) [90, 19, 69, 38, 39, 73], which operate with different levels of implicitness in different regions of the domain. This permits to apply the right scheme required to guarantee a stable solution. Another class of advanced schemes for the time complexity is the class of multirate methods. They allow for a flexible selection of time step size within the domain. In the early developments of multirate methods, the different time steps employed to integrate in time the sytem were selected a-priori, exploiting the knowledge of the problem at hand [3]. More recent extensions include a self-adjusting strategy to select the fast components in the system automatically [86]. Note that most of the developments within the multirate literature address only ODE systems and hyperbolic wave equations [20, 37]. Furthermore, they mainly implement non-conservative procedures, which can lead to stability issues when applied to coupled flow-transport systems.

The aim of this thesis is the development of advanced techniques for the space-time discretization of multiphase flows in highly heterogeneous porous media simulation. These techniques are based on the adaptation of the computational grids, in space and in time, in order to optimize the trade-off between the accuracy of the solution and the efficiency of the simulation. We developed a new multirate technique able to preserve the mass conservation based on a flux partition strategy. Local mass conservation is a desired property for numerical methods that solve time dependent conservation laws to converge to the right weak solution [62]. For the first time, we integrated this method into nonlinear multiphase flow simulators. We combined the time adaptivity, obtained by the multirate method, with different advanced techniques able to treat the spatial heterogeneity. First, we combined the multirate method with a multiscale approach. It solves the problem on a coarse grid capturing all the fine scale properties with the help of some ad hoc basis functions. Then, we integrated the multirate approach with an algebraic dynamic multilevel method for the solution of the transport equations. The obtained method is able to use different grid resolutions, both in space and time, based on an error estimator. In this way, the method employs the highest grid resolution only at the location of the moving saturation fronts.

The thesis is organized as follows:

- Chapter 2 presents a mass conservative multirate method for hyperbolic conservation laws. The properties of the method – e.g. consistency, stability, mass conservation – are analyzed in details.

- Chapter 3 illustrates the mathematical model which describes multiphase flow in porous media with a brief overview on the classical numerical schemes and strategies used to compute the numerical solution.

- Chapter 4 presents a novel conservative multirate multiscale method for multiphase flow simulation in highly heterogeneous porous media developed in collaboration with Prof. Hadi Hajibeygi of TU Delft.

- Chapter 5 shows the algebraic dynamic multilevel approach combined with a mass conservative local time-stepping strategy for the solution of the transport equation of multiphase flow in porous media developed in collaboration with Prof. Hadi Hajibeygi and Dr. Matteo Cusini of TU Delft.
- Chapter 6 contains general conclusions and some comments about future developments of the work presented in this thesis.

# A conservative implicit multirate method for hyperbolic equations

2

Multirate methods have been mostly applied to general systems of ODEs. In this chapter we focus exclusively on systems that result from the space discretization of hyperbolic conservation laws. Unlike previous attempts, we propose a component partitioning strategy which is based on the the numerical fluxes, in order to preserve the mass conservation properties of the single rate method. This approach is inspired by the flux partitioning strategy proposed in [57] to derive monotonic methods for space discretized conservation laws. First an overview of a self adjusting multirate approach is provided. Then, the proposed conservative multirate method is described in details, analyzing its properties. Several numerical experiments have been performed to test the accuracy of the solution and the efficiency of the method. In all the test cases, the TR-BDF2 time integration scheme has been used.

## 2.1 State of art

Multirate methods were originally proposed in [83] in the context of systems of ordinary differential equations. Many studies have been then devoted to the improvement of these methods, see e.g. [3], [40]. The main idea of multirate methods is to integrate each component of the system using a different time steps. Slow components – i.e. components with longer characteristic time scales – are integrated with larger time steps, while smaller time steps are used for fast components. Thus, multirate methods can avoid a significant amount of the computations compared with single rate approaches, if the faster components are confined in a small part of the domain (possibly evolving in time). In other words, in the multirate approach the most appropriate time resolution is employed for each variable of the system. In earlier multirate methods, as already mentioned in Chapter 1, the system was partitioned *a priori*, based on the knowledge of the specific problem to be solved. A self-adjusting, recursive time stepping strategy has been then proposed in [86]. In this more recent approach, a tentative global step is first taken for all components, using a robust, unconditionally stable method. The time step is then reduced only for those components for which a suitable local error estimator is greater than the specified tolerance. In this way, automatic detection of fast components is achieved.

In [20] and [37] the authors propose multirate Runge-Kutta methods that preserve the stability properties of the single rate approach. We will base our work on the strategy proposed in [86] for the $\theta$-method and extended in [9] to the TR-BDF2 method as fundamental single rate solver.

Parts of this chapter have been published in the Computational Geosciences journal [28]

While multirate methods have been mostly applied to general systems of ODEs, in this work we will focus exclusively on systems that arise from the space discretization of conservation laws. Unlike previous attempts, we propose a component partitioning strategy which is based on the the numerical fluxes, in order to preserve the mass conservation properties of the single rate method.

## 2.2 A self-adjusting multirate approach

In this section the self-adjusting multirate approach proposed in [9] is outlined, as applied to the solution of the Cauchy initial value problem

$$y'(t) = f(t, y(t)), \quad t \in (0, T], \quad y(0) = y_0 \in \mathbb{R}^m. \tag{2.1}$$

where $f(t, y) : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$ is continuous with respect to both arguments and it is Lipschitz-continuous with respect to its second argument. We consider time discretizations associated to discrete time levels $t^n,\ n = 0, \ldots, N$ such that $\Delta t_n = t^{n+1} - t^n$ and we will denote by $u^n$ the numerical approximation of $y(t^n)$. We will also denote by $u^{n+1} = \mathcal{S}(u^n, \Delta t_n)$ the implicitly defined operator $\mathcal{S} : \mathbb{R}^m \to \mathbb{R}^m$ whose application is equivalent to the computation of one step of size $\Delta t_n$ of a given single step method. While here only implicit methods will be considered, the whole framework can be extended to explicit and IMEX methods.

Let $P$ the orthogonal projector onto a linear subspace $\mathcal{A}_C \subset \mathbb{R}^m$ with dimension $p < m$, and let $\mathcal{S}^{\mathcal{A}_C} : \mathbb{R}^p \times \mathbb{R}^{m-p} \to \mathbb{R}^p$ the operator that represents the solution of the subsystem obtained by freezing the components of the unknown belonging to $\mathcal{A}_C^{\perp}$ to a value $z \in \mathbb{R}^{m-p}$. Then, the operator $\mathcal{S}^{\mathcal{A}_C}$ can be defined by $y = \mathcal{S}^{\mathcal{A}_C}(x, z, \Delta t_n) = P\mathcal{S}(x \oplus z, \Delta t_n)$. Furthermore, we will denote by $Q(u^{n+1}, u^n, \zeta)$ the interpolation operator that provides an approximation of the numerical solution at intermediate time levels $t_n + \zeta$, where $\zeta \in [0, \Delta t_n]$. Linear interpolation is often employed, but, for multistage methods, knowledge of the intermediate stages also allows the application of more accurate interpolation procedures without substantially increasing the computational cost.

In a multirate approach, system (2.1) is partitioned into a sub-system of so called *active components* with a faster time scale and into the complementary sub-system of the *latent components*, which are associated to slower phenomena. In this context, the basic idea of a self-adjusting strategy is to use a tentative global time step to identify the set of the active components, which have to be recomputed with a smaller time step to maintain the desired accuracy and stability. In particular, the self-adjusting multirate algorithm introduced in [9] is a generalization of that proposed in [86] and can be summarized as in Algorithm 1.

A stability analysis of the above described approach has been proposed in [9] in the case of a linear system with a simplified refinement strategy. The effectiveness of the above procedure depends in a crucial way on the accuracy and stability of the basic ODE solver $\mathcal{S}$, as well as on the time step

**Algorithm 1: Self-adjusting multirate**

Perform a tentative **global (or macro) time step** of size $\Delta t_n$ with the standard single rate method and compute $\hat{u}^{n+1} = \mathcal{S}(u^n, \Delta t_n)$;

Apply an error estimator to partition the state space into active and latent variables.

Set $P_n^{(0)}$, the projection onto the subspace $\mathcal{A}_{C_0}$ of the active variables.

Set $\bar{P}_n^{(0)}$, the subspace projection onto the complementary;

Set $\bar{P}_n^{(0)} u^{n+1} = \bar{P}_n^{(0)} \hat{u}^{n+1}$ $u^{n,0} = u^n$, $t^{n,0} = t^n$, and $k = 0$.;

**repeat**

- Set $k = k + 1$ ;
- Choose a local (or micro) time step $\Delta t_n^{(k)}$ for the active variables, based on the value of the error estimator ;
- Set $t^{n,k} = \min\{t^{n,k-1} + \Delta t_n^{(k)}, t^{n+1}\}$ ;
- Update the latent variables by interpolation $\bar{P}_n^{(k)} u^{n,k} = Q(\bar{P}_n^{(k-1)} u^{n+1}, \bar{P}_n^{(k-1)} u^{n,k}, \Delta t_n^{(k)})$ ;
- Update the active variables by computing $P_n^{(k)} u^{n,k} = \mathcal{S}^{\mathcal{A}_{C_{k-1}}}(P_n^{(k-1)} u^{n,k}, \bar{P}_n^{(k-1)} u^{n,k-1}, \Delta t_n^{(k)})$;
- Compute the error estimator for the active variables in $\mathcal{A}_{C_{k-1}}$ and partition again $\mathcal{A}_{C_{k-1}}$ into new latent and active variables. Denote by $\mathcal{A}_{C_k} \subset \mathcal{A}_{C_{k-1}}$ the new subspace of active variables and by $P_n^{(k)}$ the corresponding projection ;

**until** $t^{n,k} = t^{n+1}$;

Go to the next time step ;

refinement and partitioning criterion. In [9], the embedded error estimator of the TR-BDF2 method was used for the error estimator and the error control strategy proposed in [37] was extended to employ a combination of absolute and relative error tolerances. It is important to remark that the previously defined approach, when applied to ODE systems stemming from the space discretization of conservation laws like (2.2), does not guarantee mass conservation for the numerical solution, since some of the fluxes are recomputed during refinement only for one of the two adjacent variables. For this reason, in section 2.3 we propose a conservative version of the method.

## 2.3 The conservative implicit multirate approach for hyperbolic conservation laws

The aim of this section is to introduce a novel mass conservative, implicit multirate scheme to integrate in time nonlinear conservation laws of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \qquad x \in \mathbb{R}, \quad t > 0,$$

with given initial datum $u(x,0) = u_0(x)$ for $x \in \mathbb{R}$. For the sake of simplicity, in this section we will treat scalar problems only, and consider the differential problem on the whole real line, postponing to a later stage a discussion on how to treat boundary conditions for problems in a bounded domain. To discretize the equation in space, we consider the set of the cells $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, for $i \in \mathbb{Z}$, with $x_i$ being the center of cell $I_i$ and $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ the cell size.

We denote by $u_i(t)$ the approximation of the average of $u(x,t)$ in cell $I_i$, i.e $U_i(t) \simeq \dfrac{1}{\Delta x_i}\displaystyle\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x,t)\,\mathrm{d}x$ for $t > 0$, while the initial value at $t = 0$ is obtained from the initial data,

$$U_i(0) = \frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_0(x)\,\mathrm{d}x.$$

A conservative finite volume discretization yields the following system of ordinary differential equations

$$\frac{dU_i}{dt}(t) = -\frac{1}{\Delta x_i}\left[F_{i+\frac{1}{2}}(t) - F_{i-\frac{1}{2}}(t)\right], \quad i \in \mathbb{Z}, \quad t > 0, \tag{2.2}$$

where $F_{i\pm\frac{1}{2}}(t) = F(U_{i\mp p}(t), \cdots, U_i(i), \cdots, U_{i\pm q}(i))$ is the semi-discrete numerical flux at the control volume face $x_{i\pm\frac{1}{2}}$ and $x_{i\mp p}, \cdots, x_i, \cdots, x_{i\pm q}$ is the stencil of nodes used to evaluate it. For instance, in the classical two-point flux approximation $p = 0$ and $q = 1$.

Equations in the form (2.2) are the starting point for our multirate approach, which, differently from the scheme outlined in the previous section,

employs an error estimator based on the fluxes rather than on the system components to identify active and latent components, with the aim to maintain the mass conservation properties of the basic scheme.

We give here a general overview of the method, postponing to a later section a more detailed description of the algorithm. Given the numerical solution at time $t^n$ and a global time step $\Delta t_n^{(0)} = t^{n+1} - t^n$, we aim at a numerical scheme that can be written eventually in the form

$$u_i^{n+1} = u_i^n - \frac{1}{\Delta x_i}\left(H_{i+\frac{1}{2}} - H_{1-\frac{1}{2}}\right), \tag{2.3}$$

where

$$H_{i\pm\frac{1}{2}} \cong \int_{t^n}^{t^{n+1}} F_{i\pm\frac{1}{2}} \mathrm{d}t$$

is the numerical flux, which typically depends on $F_{i\pm\frac{1}{2}}$ sampled at different times. Note that we are using a non-standard definition for the numerical flux, since we are not dividing the time integral by the time step length. Furthermore, we have not yet indicated the time at which the numerical fluxes are evaluated, since we will consider both implicit and explicit formulations. Discretizations in the form (2.3) are conservative in the sense that, for any set of indices $\mathcal{I}$, the quantity $\sum_{i\in\mathcal{I}} \Delta x_i (u_i^{n+1} - u_i^n)$ depends only on the values of the numerical fluxes at the boundary of the set $\cup_{i\in\mathcal{I}} I_i$.

At each time step, we first compute the approximate solution at $t^{n+1}$ for all components with a tentative time step. The accuracy of the numerical fluxes at all interfaces is checked using an appropriate error estimator. If the flux is rejected on the basis of the error estimator, all components involved in its stencil are added to the set of active components that need to be recomputed with a smaller time step. During the re-computation, the accepted numerical fluxes are kept constant inside the time slab and interpolation in time is used to obtain their appropriate value, while the rejected ones are recomputed. In this way, interpolation is applied directly to the fluxes, rather than to the components, which allows to maintain the structure of the scheme in the form (2.3), where the $H_{i\pm\frac{1}{2}}$ will consist, at the end of the procedure, of contributions coming from the accepted fluxes.

### 2.3.1 A first example

For the sake of clarity, we first present the proposed multirate method using the $\theta$-method as implicit time integration scheme, and we use a uniform grid of size $\Delta x$. The purpose is to introduce the scheme on a simple example before presenting its most general description. We will also employ a two-point flux approximation, which means that $F_{i\pm\frac{1}{2}} = F(u_i, u_{i\pm1})$ . At the global time level $t^{n,(0)}$, using the time step $\Delta t_n = \Delta t$, the following expression is obtained in the first tentative calculation:

$$\hat{u}_i^{n+1} = u_i^n - \frac{\theta}{\Delta x}\left[F_{i+\frac{1}{2}}^{n+1} - F_{i-\frac{1}{2}}^{n+1}\right] - \frac{1-\theta}{\Delta x}\left[F_{i+\frac{1}{2}}^{n} - F_{i-\frac{1}{2}}^{n}\right],$$

where $F^n_{1\pm\frac{1}{2}}$ denotes the numerical flux computed using the value of the approximated components at time $t^n$. Clearly, with a simple reformulation the scheme can be rewritten in the form (2.3). We recall that $\Delta t_n$ is included in the numerical fluxes, in contrast to the usual definitions found in the literature.

If we suppose, as shown in Fig. 2.1, that the error estimator rejects the flux at the interface point $x_{i+\frac{1}{2}}$, we have to recompute the components in the stencil of $F_{i+\frac{1}{2}}$. Therefore, $u_i$ and $u_{i+1}$ will be recomputed using a smaller time step. Here, for simplicity, we reduce $\Delta t_n^{(0)}$ by a half. If instead $F_{i-\frac{1}{2}}$ is accepted, at the new intermediate time $t^{n+\frac{1}{2}} = t^n + \Delta t_n^{(1)} = t^n + \frac{1}{2}\Delta t_n^{(0)}$ we have

$$u_i^{n+\frac{1}{2}} = u_i^n - \frac{\theta}{\Delta x}\left[F^{n+\frac{1}{2}}_{i+\frac{1}{2}} - \frac{1}{2}F^{n+1}_{i-\frac{1}{2}}\right] - \frac{1-\theta}{\Delta x}\left[F^n_{i+\frac{1}{2}} - \frac{1}{2}F^n_{i-\frac{1}{2}}\right].$$

Here, $F^n_{1-\frac{1}{2}}$ and $F^{n+1}_{1-\frac{1}{2}}$ have been kept frozen at the value computed at the larger time step (since $F_{i-\frac{1}{2}}$ has been accepted). They are multiplied by a factor $\frac{1}{2}$ to account for the time step reduction $\frac{\Delta t_n^{(1)}}{\Delta t_n^{(0)}}$. As for cell $i+1$, if we suppose to accept the numerical flux at the interface point $x_{i+\frac{3}{2}}$, a similar expression is obtained:

$$u_{i+1}^{n+\frac{1}{2}} = u_{i+1}^n - \frac{\theta}{\Delta x}\left[\frac{1}{2}F^{n+1}_{i+\frac{3}{2}} - F^{n+\frac{1}{2}}_{i+\frac{1}{2}}\right] - \frac{1-\theta}{\Delta x}\left[\frac{1}{2}F^n_{i+\frac{3}{2}} - F^n_{i+\frac{1}{2}}\right].$$

If the new time step $\Delta t_n^{(1)}$ is such that all fluxes are accepted, we can recompute the solution at time $t^{n+1}$ as

$$u_i^{n+1} = u_i^{n+\frac{1}{2}} - \frac{\theta}{\Delta x}\left[F^{n+1}_{i+\frac{1}{2}} - \frac{1}{2}F^{n+1}_{i-\frac{1}{2}}\right] - \frac{1-\theta}{\Delta x}\left[F^{n+\frac{1}{2}}_{i+\frac{1}{2}} - \frac{1}{2}F^n_{i-\frac{1}{2}}\right],$$
$$u_{i+1}^{n+1} = u_{i+1}^{n+\frac{1}{2}} - \frac{\theta}{\Delta x}\left[\frac{1}{2}F^{n+1}_{i+\frac{3}{2}} - F^{n+1}_{i+\frac{1}{2}}\right] - \frac{1-\theta}{\Delta x}\left[\frac{1}{2}F^n_{i+\frac{3}{2}} - F^{n+\frac{1}{2}}_{i+\frac{1}{2}}\right].$$

For cell $i-1$, if also the flux $F_{i-\frac{3}{2}}$ has been accepted, the solution at time $t^{n+1}$ is simply:

$$u_{i-1}^{n+1} = u_i^n - \frac{\theta}{\Delta x}\left[F^{n+1}_{i-\frac{1}{2}} - F^{n+1}_{i-\frac{3}{2}}\right] - \frac{1-\theta}{\Delta x}\left[F^n_{i-\frac{1}{2}} - F^n_{i-\frac{3}{2}}\right].$$

One can verify that mass conservation at the global step is guaranteed, since all fluxes at interface $i+\frac{1}{2}$ and $i-\frac{1}{2}$ cancel each other exactly. Since the choice of $i$ is arbitrary, this fact holds true for all interfaces.

### 2.3.2 The time refinement and time stepping strategy

We now present the general algorithm to perform numerical integration inside one global step $t^n \to t^{n+1}$. The algorithm is recursive and, inside the global step, we define a new sub-step each time a flux has been rejected

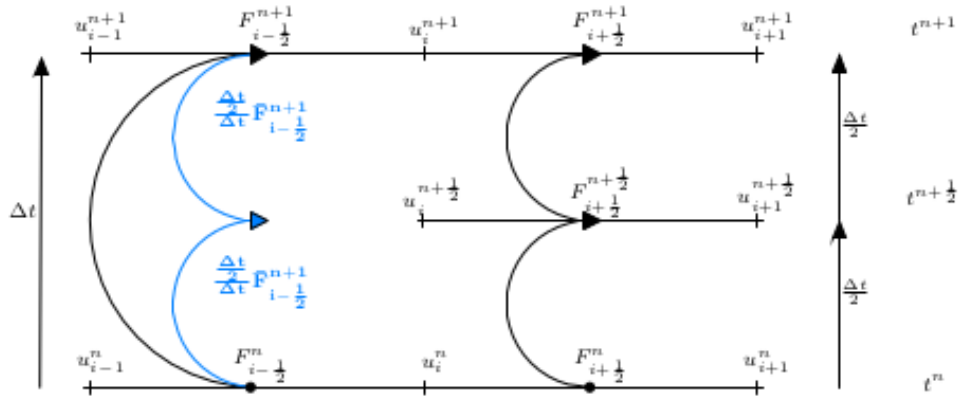

**Fig. 2.1.:** An example of flux partitioning that preserves mass at each global time step.

at the current sub-step. Moreover, in the general case the refinement ratio can be different from $\frac{1}{2}$. We will indicate with $\mathcal{A}_C$ and $\mathcal{A}_F$ the set of active components (i.e. those that have to be recomputed) and of accepted fluxes, respectively. Superscripts may be added to indicate different instances. These sets always satisfy the property

$$\mathcal{A}_C = \{u_i : \ F_{i-\frac{1}{2}} \in \mathcal{R}_F \vee F_{i+\frac{1}{2}} \in \mathcal{R}_F\}$$

where $\mathcal{R}_F$ is the set of the rejected fluxes ($\mathcal{A}_F^{\perp}$). The set of the fluxes, accepted and rejected, involved to compute the active components is called $\mathcal{F} = \{ F_{i\pm\frac{1}{2}} : u_i \in \mathcal{A}_C\}$. We also introduce the vector $\mathcal{T}_F$, that for each flux in $\mathcal{A}_F$ records the length of the sub-step when the flux has been accepted. For consistency of notation, we will use subscripts of the form $i \pm \frac{1}{2}$ to indicate fluxes or flux related quantities. We assume that an error estimator for the fluxes is provided and we only consider a two-point flux approximation, although the procedure can be extended to other types of numerical flux constructions. We denote by $S$ the operator that starting from $u^*$ returns the vector $u^{\triangle}$ of updated active components within a given sub-step, and also computes the new sets $\mathcal{A}_F$ and $\mathcal{A}_C$, together with the new time step to be used for the refined sub-steps or the next step. Operator $S$, described in Algorithm 3 , is the building block for the operator $M$, which is described in Algorithm 2 . The operator $M$ is used recursively to compute a single global time step with our multirate method. It takes as input a set of components $u^*$ and a time step, proceeding then recursively across all rejected sub-steps to produce the final value at the end of the time step. The parameter $p$ takes track of the level of refinement.

The first time that the multirate algorithm is applied so, the algorithm M is called, $p$ will be equal to $0$, $u^* = u^n$, $u^{\triangle} = \hat{u}^{n+1}$ and $\Delta t^* = \Delta t^{\triangle} = \Delta t_n$. Time $t^*$ and $t^{\triangle}$ are the times where $u^*$ and $u^{\triangle}$ have been computed, respectively.

In Algorithm 2, the index $p$, as said before, indicates the level of refinement, while the index $s$ is the sub-step taken at each level of refinement.

**Algorithm 2: Operator M**

**input** : $u^*, \Delta t^*, p$

**output** : $u^{\triangle}, \Delta t^{\triangle}$

Set $s = 1$ ;
**while** $t^* + \Delta t^* \leq t^{\triangle}$ **do**
    $u^{(s)} = u^*$;
    **if** $p = 0$ **then**
        Set $\mathcal{A}_C^{(0)}$ equal to the set of all components, $\mathcal{A}_F^{(0)} = \emptyset$, $\mathcal{T}_F^{(0)} = \emptyset$ ;
    **end**
    Call $S(u^*, \mathcal{A}_C^{(p)}, \mathcal{A}_F^{(p)}, \mathcal{T}_F^{(p)}, \Delta t^*; u^{\triangle}, \mathcal{A}_C^{(p+1)}, \mathcal{A}_F^{(p+1)}, \mathcal{T}_F^{(p+1)}, \Delta t^{\triangle})$;
    **if** $\mathcal{A}_C^{(p+1)} \neq \emptyset$ **then**
        Call $M(u^{(s)}, \Delta t^{\triangle}, p+1; u^{(s+1)}; \Delta t^{(s+1)})$ ;
    **else**
        Set $u^* = u^{\triangle}$, $t^* = t^{\triangle}$ and $\Delta t^* = \Delta t^{\triangle}$;
        set $s = s + 1$ ;
    **end**
**end**

Note that the set of fluxes marked as accepted at the given level are kept as such on all sub-steps associated to that level. This is the key for mass conservation, as explained later.

**Algorithm 3: Operator S**

**input** : $u^*, \mathcal{A}_C^*, \mathcal{A}_F^*, \mathcal{T}_F^*, \Delta t^*$

**output**: $u^\triangle, \mathcal{A}_C^\triangle, \mathcal{A}_F^\triangle, \mathcal{T}_F^\triangle, \Delta t^\triangle$

Compute $u^\triangle$ $\forall$ components in $\mathcal{A}_C^*$ starting from $u^*$ and using $\Delta t^*$:

$$u_i^\triangle = u_i^* + \sum_{i\pm\frac{1}{2}\in\mathcal{A}_F} \frac{\Delta t^*}{\Delta t_{i\pm\frac{1}{2}}} F_{i\pm\frac{1}{2}} + \sum_{i\pm\frac{1}{2}\in\mathcal{R}_F} F_{i\pm\frac{1}{2}}$$

where $\mathcal{R}_F = \mathcal{F} \setminus \mathcal{A}_C$ is the set of rejected fluxes. $\Delta t_{i\pm\frac{1}{2}}$ indicates the corresponding elements of $\mathcal{T}_F^*$ ;

Estimate the error $e_{i+\frac{1}{2}}$ for all the rejected fluxes $\mathcal{R}_F$ ;

Compute the set of rejected flues for the new sub-step:

$\mathcal{R}_F = \{F_{i+\frac{1}{2}} : e_{i+\frac{1}{2}} > \text{tol}\}$ ;

**if** $\mathcal{R}_F \neq \emptyset$ **then**

  Compute the set of active components for the new sub-step

  $\mathcal{A}_C^\triangle = \{u_i : \ F_{i-\frac{1}{2}} \in \mathcal{R}_F \vee F_{i+\frac{1}{2}} \in \mathcal{R}_F\}$ ;

  Compute the new time step for the next sub-step

$$\Delta t^{new} = \nu \min_{F_{i+\frac{1}{2}}\in\mathcal{R}_F} \left( \frac{\tau_r |F_{i+\frac{1}{2}}| + \tau_a}{\epsilon_{i+\frac{1}{2}}} \right)^{\frac{1}{r+1}} ; \qquad (2.4)$$

  Set $\Delta t^\triangle = \min\{\Delta t^*, \Delta t^{new}\}$ ;

**else**

  Set $\mathcal{A}_C^\triangle = \emptyset$ and $\Delta t^\triangle = \Delta t^*$ ;

**end**

The formula in Algorithm 3 for the new time step (2.4) is an extension of the formula originally proposed in [37] and already adapted in [9]. In (2.4), $\tau_r$ and $\tau_a$ are a relative and absolute tolerance, respectively, $r$ is the order of convergence of the chosen time advancing method and $\nu$ an user defined parameter taking values in $(0, 1]$. As customary in adaptive time integration approaches, see e.g. [80], these parameters are employed to tune the adaptation criterion and to impose a more conservative choice of the time step if necessary.

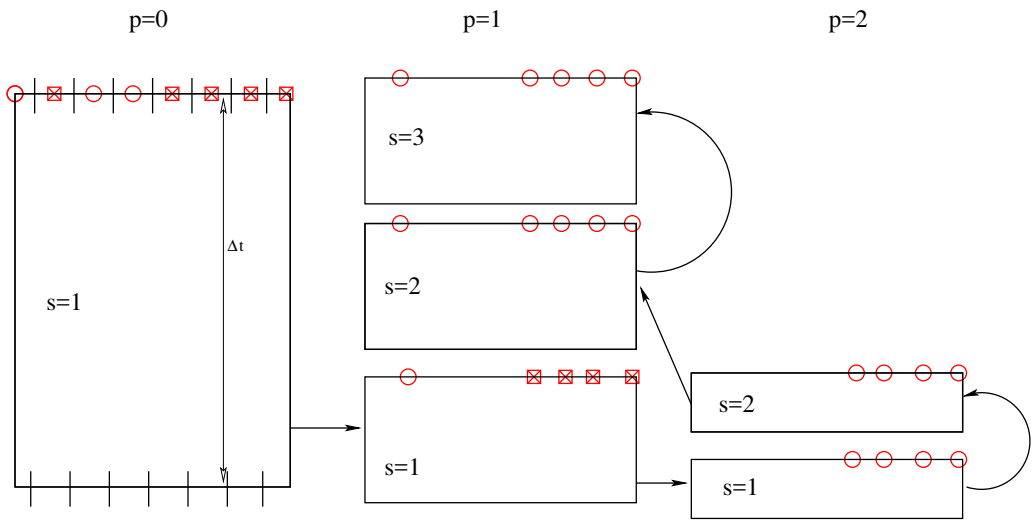


**Fig. 2.2.:** Example of the time stepping multirate method.

We mention that the Algorithm 3 keeps track also of the time instants the fluxes have to be computed, for the sake of simplicity we have omitted to indicate it explicitly. In Fig. 2.2 we sketch an example of what is obtained combining the two algorithms. The circles indicate the latent components inside the sub-step, while the crosses indicate the active components that have to be recomputed in the next sub-refinement step.

## 2.4 Properties: Mass conservation, Consistency, Stability

### 2.4.1 Mass conservation

Considering a global step $\Delta t_n = t^{n+1} - t^n$, the values of the numerical approximation using the multirate approach and the $\theta$-method as time-advances scheme, can be written in the following way:

$$u_i^{n+1} = u_i^n - \frac{1}{\Delta x}\left[H^+_{i+\frac{1}{2}} - H^-_{i-\frac{1}{2}}\right] \tag{2.5}$$

where the fluxes are

$$H^+_{i+\frac{1}{2}} = \sum_{p,s:\{u_i^{p,s}\in\mathcal{A}_C^{p,s*}\wedge u_i^{p,s}\notin\mathcal{A}_C^{p,s\triangle}\}} \left\{\theta H^{p,s\triangle}_{i+\frac{1}{2}} + (1-\theta)H^{p,s*}_{i+\frac{1}{2}}\right\}$$

$$H^-_{i-\frac{1}{2}} = \sum_{p,s:\{u_i^{p,s}\in\mathcal{A}_C^{p,s*}\wedge u_i^{p,s}\notin\mathcal{A}_C^{p,s\triangle}\}} \left\{\theta H^{p,s\triangle}_{i-\frac{1}{2}} + (1-\theta)H^{p,s*}_{i-\frac{1}{2}}\right\}$$

with:

$$H^{p,s\triangle}_{i+\frac{1}{2}} = \begin{cases} F^{p,s,\triangle}_{i+\frac{1}{2}} & \text{if } F^{p,s}_{i+\frac{1}{2}} \notin \mathcal{A}_F^{p,s*} \\ \dfrac{\Delta t^{p,s}}{\Delta t^{\hat{p},\hat{s}}} F^{\hat{p},\hat{s}\triangle}_{i+\frac{1}{2}} & \text{otherwise} \end{cases},$$

similarly,

$$H_{i+\frac{1}{2}}^{p,s*} = \begin{cases} F_{i+\frac{1}{2}}^{p,s,\triangle} & \text{if } F_{i+\frac{1}{2}}^{p,s} \notin \mathcal{A}_F^{p,s*} \\ \\ \dfrac{\Delta t^{p,s}}{\Delta t^{\hat{p},\hat{s}}} F_{i+\frac{1}{2}}^{\hat{p},\hat{s}*} & \text{otherwise} \end{cases},$$

$\hat{s}$ and $\hat{p}$ denote the sub-steps of the previous sub-refinement where the flux has been accepted (the last $p, s : F_{i+\frac{1}{2}}^{p,s} \in \mathcal{A}_F^{p,s*}$)

**Proposition 2.4.1.** *Let $i$ and $i-1$ two generic consecutive cells, computed with the method* (2.5)*, then the fluxes at the common interface $H_{i-\frac{1}{2}}^{-}$ and $H_{i-\frac{1}{2}}^{+}$, are equal.*

*Proof.* It is easy to show that $H_{i-\frac{1}{2}}^{+} = H_{i-\frac{1}{2}}^{-}$ if:

$$\forall p, s : \{u_i^{s,p} \in \mathcal{A}_C^{p,s*} \wedge u_i^{p,s} \notin \mathcal{A}_C^{p,s\triangle}\} \iff \{u_{i-1}^{s,p} \in \mathcal{A}_C^{p,s*} \wedge u_{i-1}^{p,s} \notin \mathcal{A}_C^{p,s\triangle}\}$$

because in this case both cells $i$ and $i-1$ have been accepted in the same sub-step and the number of evaluated fluxes as their values are the same.

Instead if,

$$\exists \bar{p}, \bar{s} : \{u_{i-1}^{\bar{p},\bar{s}} \in \mathcal{A}_C^{\bar{p},\bar{s}*} \wedge u_{i-1}^{\bar{p},\bar{s}} \notin \mathcal{A}_C^{\bar{p},\bar{s}\triangle}\} \wedge \{u_i^{\bar{p},\bar{s}} \in \mathcal{A}_C^{\bar{p},\bar{s}*} \wedge u_i^{\bar{p},\bar{s}} \in \mathcal{A}_C^{\bar{p},\bar{s}\triangle}\}$$

this means that the flux $F_{i-\frac{1}{2}}$ in that step has been accepted, because the component $u_{i-1}$ is not active, but the flux $F_{i+\frac{1}{2}}$ has been rejected and it has to be recomputed, so that a new sub-refinement is required.

The flux $H_{i-\frac{1}{2}}^{+}$ at this point can be written as:

$$H_{i-\frac{1}{2}}^{+} = \sum_{p,s}^{\bar{p},\bar{s}-1} \left(\theta H_{i-\frac{1}{2}}^{p,s\triangle} + (1-\theta) H_{i-\frac{1}{2}}^{p,s*}\right) + \theta F_{i-\frac{1}{2}}^{\bar{p},\bar{s}\triangle} + (1-\theta) F_{i-\frac{1}{2}}^{\bar{p},\bar{s}*}$$

and $H_{i-\frac{1}{2}}^{-}$ as:

$$H_{i-\frac{1}{2}}^{-} = \sum_{p,s}^{\bar{p},\bar{s}-1} \left(\theta H_{i-\frac{1}{2}}^{p,s\triangle} + (1-\theta) H_{i-\frac{1}{2}}^{p,s*}\right) + \sum_{j=1}^{N} \theta \frac{\Delta t^j}{\Delta t^{\bar{p},\bar{s}}} F_{i-\frac{1}{2}}^{\bar{p},\bar{s}\triangle} + \sum_{j=1}^{N} (1-\theta) \frac{\Delta t^j}{\Delta t^{\bar{p},\bar{s}}} F_{i-\frac{1}{2}}^{\bar{p},\bar{s}*},$$

the $j$ steps are all the later steps of the later sub - refinements. Due to the recursive nature of the algorithm, we have that $\sum_{j=1}^{N} \Delta t^j = \Delta t^{\bar{p},\bar{s}}$ because the Algorithm 2 ends when the synchronization between two consecutive sub-refinements is verified. So that the two different contribution at the end have the same value $H_{i-\frac{1}{2}}^{+} = H_{i-\frac{1}{2}}^{-}$.

This argument is easily applicable also in the opposite case, when $u_i$ is a not active component while $u_{i-i}$ is an active component

$\left(\{u_i^{\hat{s},\hat{p}} \in \mathcal{A}_C^{\hat{p},\hat{s}*} \wedge u_i^{\hat{p},\hat{s}} \notin \mathcal{A}_C^{\hat{p},\hat{s}\triangle}\} \wedge \{u_{i-1}^{\hat{s},\hat{p}} \in \mathcal{A}_C^{\hat{p},\hat{s}*} \wedge u_{i-1}^{\hat{p},\hat{s}} \in \mathcal{A}_C^{\hat{p},\hat{s}\triangle}\}\right)$.

Since there are no other possible cases, the correct flux balance is preserved at each interface of the domain for each global time steps. □

**Remark 2.4.1.** *Since the fluxes at the interface between two consecutive generic cells are the same this means that the local conservation mass is preserved at the and of each global time step.*

### 2.4.2 Consistency

In [53] the authors analyze not self-adjusting explicit multirate schemes of Runge-Kutta type applied to hyperbolic equations and show that a scheme of that sort is either consistent or conservative.

In this section we will recall the concept of consistency of a numerical scheme for hyperbolic equations, by introducing a definition of weak consistency more suited for finite volume schemes of the type considered in this work. We then recover the results of [53] for the classic consistency concept used in finite differences, showing that weak consistency is instead maintained.

We finally show that the application of self-adjusting multirate strategy allows to recover consistency in both senses. We start to recall the problem we wish to approximate,

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad x \in \mathbb{R},\ t > 0, \tag{2.6}$$

for a given initial data $u_0$. Here $f(u)$ is the flux, assumed to be a continuosly differentiable function. We also assume that the problem is well-posed and $u$ is bounded.

Given a spatial grid of size $\Delta x$ and a time-step size $\Delta t$ we indicate with

$$U_i^n = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(t_n, x)\, dx,$$

the average solution on cell $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$ at time $t^n$, and with

$$\underline{F}_{i\pm\frac{1}{2}} = \int_{t_n}^{t_{n+1}} f\left(u\left(t, x_{i\pm\frac{1}{2}}\right)\right)\, dt \tag{2.7}$$

the integrated exact flux at location $x_{i\pm 1/2}$.

We consider discretizations of (2.6) of the form

$$u_i^{n+1} - u_i^n = \frac{1}{\Delta x}\left(F^\theta_{i-\frac{1}{2}} - F^\theta_{i+\frac{1}{2}}\right), \tag{2.8}$$

where $\theta = n$ or $\theta = n+1$ depending on the choice of an explicit or implicit scheme, respectively,

$$F^\theta_{i+\frac{1}{2}} = \Delta t \widehat{F}(u_i^\theta, u_{i+1}^\theta) \simeq \underline{F}_{i+\frac{1}{2}}$$

is a two-point flux approximation, while $u_i^n$ is the approximation of $U_i^n$. The initial data is given by $u_i^0 = U_i^0$. In this section we indicate with $\widehat{F}(a, b)$ the numerical flux function that characterize the specific scheme, to avoid

confusion with the numerical flux $F_{i+\frac{1}{2}}$, which in our notation depends also on $\Delta t$.

We provide the following definition, see [11, 35].

**Definition 2.4.1.** *A finite volume scheme in the form* (2.8) *is consistent if the numerical flux function $\widehat{F}$ is Lipschitz continuous in its arguments and satisfies the following consistency condition,*

$$\widehat{F}(w, w) = f(w), \quad \forall w \in \mathbb{R},$$

The previous definition implies the following

**Proposition 2.4.2.** *For a scheme of the form* (2.8) *with fluxes satisfying the conditions in Definition 2.4.1 we have that*

$$u_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(\delta_{i+\frac{1}{2}} - \delta_{i-\frac{1}{2}}\right), \quad \forall i,\ \forall n \geq 0, \tag{2.9}$$

*with $\lim_{\Delta t, \Delta x \to 0} \delta_{i\pm\frac{1}{2}} = 0$, when setting $u_i^n = U_i^n$ and $F^\theta_{i\pm\frac{1}{2}} = \Delta t \widehat{F}(U_i^\theta, U_{i\pm1}^\theta)$ in scheme* (2.8)*. We will term this property as* weak consistency.

We note that (2.9) is related to weak convergence, see [11], and is different from the one classically used for finite difference schemes, based on the analysis of the local truncation error, defined as

$$\tau_i = \frac{u(t^{n+1}, x_i) - u(t^n, x_i)}{\Delta t} + \frac{\widehat{F}(u(t^\theta, x_i), u(t^\theta, x_{i+1})) - \widehat{F}(u(t^\theta, x_{i-1}), u(t^\theta, x_i))}{\Delta x}.$$

In the finite difference framework, a scheme is said to be consistent if, for all $i$, $\lim_{\Delta t, \Delta x \to 0} \tau_i = 0$. We may note that instead (2.9) is equivalent to set

$$\hat{\tau}_i = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left[\frac{u(t^{n+1}, x) - u(t^n, x)}{\Delta t} + \frac{\widehat{F}(U_i^\theta, U_{i+1}^\theta) - \widehat{F}(U_{i-1}^\theta, U_i^\theta)}{\Delta x}\right] dx$$

with $\hat{\tau}_i = \delta_{i+\frac{1}{2}} - \delta_{i-\frac{1}{2}}$, and this latter, weaker, formulation is more suitable for finite volume schemes, since in these schemes $u_i^n \simeq U_i^n$.

*Proof.* If we integrate (2.6) in space and time in the interval $I^n \times [t^n, t^{n+1}]$ we have

$$U_i^{n+1} - U_i^n + \frac{F^\theta_{i+\frac{1}{2}} - F^\theta_{i-\frac{1}{2}}}{\Delta x} = 0.$$

We take $u_i^n = U_i^n$ and $F^\theta_{i\pm\frac{1}{2}} = \Delta t \widehat{F}(U_i^\theta, U_{i\pm1}^\theta)$ in (2.8), subtracting the previous relation we obtain an expression of the form (2.9) with

$$\delta_{i\pm\frac{1}{2}} = \frac{F_{i\pm\frac{1}{2}} - F_{i\pm\frac{1}{2}}}{\Delta t}.$$

Thanks to the consistency and Lipshitz continuity of the numerical flux function we have

$$F_{i\pm\frac{1}{2}} = \Delta t \widehat{F}(U_i^\theta, U_{i\pm1}^\theta) = \Delta t \left[ f\left(u\left(t^\theta, x_{i\pm\frac{1}{2}}\right)\right) + O(\Delta x)\right],$$

while, from (2.7),

$$\underline{F}_{i\pm\frac{1}{2}} = \Delta t f\left(u\left(t^\theta, x_{i\pm\frac{1}{2}}\right)\right) + O(\Delta t^2).$$

By which, $\delta_{i\pm\frac{1}{2}} = O(\Delta t) + O(\Delta x)$, and the Proposition is proved. □

Note that we have proved at least first order of consistency, however, by exploiting the expression of a specific numerical flux, for instance the Lax-Wendroff flux, we can better the order of consistency.

We now state that

**Proposition 2.4.3.** *A non self-adjusting multirate scheme which uses* (2.8) *as basic scheme is not consistent in a finite-difference sense, while we can recover weak consistency.*

*Proof.* Short of giving a general proof, we consider the special case of an explicit-scheme. Same arguments may be used for an implicit scheme. First of all we recall that in a multirate scheme relation (2.8) is effectively replaced by

$$u_i^{n+1} - u_i^n = \frac{1}{\Delta x}\left(H_{i-\frac{1}{2}}^\theta - H_{i+\frac{1}{2}}^\theta\right),$$

where $H_{i\pm\frac{1}{2}}$ is possibly built by a linear combination of $F_{i\pm\frac{1}{2}}$ taken at different sub-steps within $[t^n, t^{n+1}]$.

We assume that just the flux $F_{i+\frac{1}{2}}^n$ has been (a-priori) rejected, and thus refined by computing its value $F_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ at an intermediate step. We have

$$\begin{aligned}
u_i^{n+1} &= u_i^{n+\frac{1}{2}} - \frac{1}{\Delta x}\left(F_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \frac{1}{2}F_{i-\frac{1}{2}}^n\right),\\
u_{i+1}^{n+1} &= u_{i+1}^{n+\frac{1}{2}} - \frac{1}{\Delta x}\left(\frac{1}{2}F_{i+\frac{3}{2}}^n - F_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right),\\
u_i^{n+\frac{1}{2}} &= u_i^n - \frac{1}{2\Delta x}\left(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n\right),\\
u_{i+1}^{n+\frac{1}{2}} &= u_{i+1}^n - \frac{1}{2\Delta x}\left(F_{i+\frac{3}{2}}^n - F_{i+\frac{1}{2}}^n\right),
\end{aligned}$$

where $F_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \dfrac{\Delta t}{2}\widehat{F}\left(u_i^{n+\frac{1}{2}}, u_{i+1}^{n+\frac{1}{2}}\right)$. Using the previous relations, we have

$$u_i^{n+1} = u_i^n - \frac{1}{\Delta x}\left[\left(\frac{1}{2}F_{i+\frac{1}{2}}^n + F_{i+\frac{1}{2}}^{n+\frac{1}{2}}\right) - F_{i-\frac{1}{2}}^n\right]$$

We can recognize that $H_{i+\frac{1}{2}}^n = \dfrac{1}{2}F_{i+\frac{1}{2}}^n + F_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ and $H_{i-\frac{1}{2}}^n = F_{i-\frac{1}{2}}^n$. We first prove that we do not satisfy finite-difference consistency. For this purpose,

we use the short hand notation $u_i^n = u(t+n, x_i)$ and $u_{i+\frac{1}{2}}^n = u\left(t+n, x_{i+\frac{1}{2}}\right)$. We also assume that $u$ be sufficiently regular and $\widehat{F}$ twice continuously differentiable. The local truncation error may be written as

$$\tau_i^n = \frac{u_i^{n+1}-u_i^n}{\Delta t} + \frac{\widehat{F}\left(u_{i+1}^n,u_i^n\right)-\widehat{F}\left(u_{i-1}^n,u_i^n\right)}{\Delta x} + \frac{\widehat{F}\left(\tilde{u}_{i+1}^{n+\frac{1}{2}},\tilde{u}_i^{n+\frac{1}{2}}\right)-\widehat{F}\left(u_{i+1}^n,u_i^n\right)}{2\Delta x}, \tag{2.10}$$

where we recall that the numerical fluxes $F_{i\pm1/2}^n$ are computed using the exact solution at location $x_{i\pm1/2}$, respectively. Here,

$$\begin{aligned}
\tilde{u}_i^{n+\frac{1}{2}} &= u_i^n - \frac{\Delta t}{2\Delta x}\left(\widehat{F}\left(u_{i+1}^n, u_i^n\right) - \widehat{F}\left(u_{i-1}^n, u_i^n\right)\right), \\
\tilde{u}_{i+1}^{n+\frac{1}{2}} &= u_{i+1}^n - \frac{\Delta t}{2\Delta x}\left(\widehat{F}\left(u_{i+2}^n, u_{i+1}^n\right) - \widehat{F}\left(u_{i+1}^n, u_i^n\right)\right).
\end{aligned}$$

The first two terms on the right-hand side of (2.10) form the basic scheme, which is by hypothesis finite-difference consistent, so they give an infinitesimal w.r.t. $\Delta t$ and $\Delta x$. It remains to examine the numerator of the last fraction.

By expanding in Taylor series and neglecting higher order terms, we can easily obtain that

$$\widehat{F}\left(\tilde{u}_{i+1}^{n+\frac{1}{2}}, \tilde{u}_i^{n+\frac{1}{2}}\right) - \widehat{F}\left(u_{i+1}^n, u_i^n\right) = O(\Delta t) + O(\Delta x),$$

by which,

$$\frac{\widehat{F}\left(\tilde{u}_{i+1}^{n+\frac{1}{2}}, \tilde{u}_i^{n+\frac{1}{2}}\right) - \widehat{F}\left(u_{i+1}^n, u_i^n\right)}{2\Delta x} = O\left(\frac{\Delta t}{\Delta x}\right).$$

If we refine at a constant Courant number $\dfrac{\Delta t}{\Delta x}$ is constant and thus we are introducing a term $O(1)$ in the local truncation error. Thus we may conclude that the scheme is not consistent (in the finite difference sense).

Now, we proceed as in Proposition 2.4.2. In particular, we have set $u_i^n = U_i^n$ and, since we are using an explicit scheme, also the numerical fluxes at time step $n$ are computed using the $U^n$ values.

We find that $\delta_{i-\frac{1}{2}}$ has the same expression as before, but instead

$$\delta_{i+\frac{1}{2}} = \frac{1}{\Delta t}\left(F_{i+\frac{1}{2}}^n - H_{i+\frac{1}{2}}^n\right),$$

with $H_{i+\frac{1}{2}}^n = \dfrac{\Delta t}{2}\left(\widehat{F}\left(U_i^n, U_{i+1}^n\right) + \widehat{F}\left(\tilde{u}_i^{n+\frac{1}{2}}, \tilde{u}_{i+1}^{n+\frac{1}{2}}\right)\right)$, where now

$$\begin{aligned}
\tilde{u}_i^{n+\frac{1}{2}} &= U_i^n - \frac{\Delta t}{2\Delta x}\left(\widehat{F}\left(U_{i+1}^n, U_i^n\right) - \widehat{F}\left(U_{i-1}^n, U_i^n\right)\right), \\
\tilde{u}_{i+1}^{n+\frac{1}{2}} &= U_{i+1}^n - \frac{\Delta t}{2\Delta x}\left(\widehat{F}\left(U_{i+2}^n, U_{i+1}^n\right) - \widehat{F}\left(U_{i+1}^n, U_i^n\right)\right).
\end{aligned}$$

It is clear that $H^n_{i+\frac{1}{2}}$ is in fact a three point numerical flux, since it depends indirectly also on $U^n_{i+2}$. However, it is still Lipschitz continuous, being the composition of Lipschitz continuous functions, and if we set $U^n_i = U^n_{i+1} = U^n_{i+2} = U$ we obtain $f(U)$, thanks to the consistency of the numerical flux functions of the basic numerical scheme. So we can apply Proposition 2.4.2 to reach the conclusion that weak consistency is still verified.

On the other hand, it is also possible to verify that

$$\widehat{F}\left(\tilde{u}_i^{n+\frac{1}{2}}, \tilde{u}_{i+1}^{n+\frac{1}{2}}\right) = F\left(U_i^n, U_i^n\right) + O(\Delta t) + O(\Delta x) = f\left(U_i^n\right) + O(\Delta x) + O(\Delta x).$$

and so, following the same arguments as in Proposition 2.4.2, we reach the conclusion that $\delta_{i\pm\frac{1}{2}} = O(\Delta t) + O(\Delta x)$ and thus the weak consistency condition is satisfied.

□

The last result prompts two considerations. The first is that the scope of a multirate scheme is to perform local refinements *in time* on a given spatial grid, in order to ensure a good accuracy in time of the discrete solution. In that respect we could consider $\Delta x$ as fixed. So consistency for $\Delta x \to 0$ may be of concern only when multirate is coupled with adaption in space, since it may hinder convergence.

The second consideration regards the self-adjusting multirate schemes proposed in this work, and is detailed in the next subsection.

### Consistency of self-adjusting multirate schemes

In self-adjusting multirate schemes the decision to accept or refine at a certain location is driven by an error indicator $e$, which is typically function of the discrete solution or fluxes. In our proposed scheme the error indicator at the interface $x_{i+\frac{1}{2}}$,denoted $e_{i+\frac{1}{2}}$, depends on the numerical fluxes computed at two different time steps. We have

**Proposition 2.4.4.** *If, given a tolerance $tol > 0$ there exists a $\Delta t^*$ and such that for all $i$ the error indicator $e_{i+\frac{1}{2}}$ satisfies $e_{i+\frac{1}{2}} > tol$ whenever $\Delta t < \Delta t^*$ and for any $\Delta x$ sufficiently small, then a self-adjusting conservative multirate scheme using a fully consistent* (2.8) *as basic scheme is consistent both in finite difference and finite volume (weak) sense.*

*Proof.* It is sufficient to recall that in our conservative multirate scheme the multirate flux is a linear combination of basic numerical fluxes possibly computed at intermediate sub-steps. In general, we may write

$$H^{\theta}_{i\pm\frac{1}{2}} = \alpha_0 F_{i\pm\frac{1}{2}} + \sum_{i=1}^{s} \alpha_s F^{n+\delta_s}_{i\pm\frac{1}{2}}$$

with $\delta_s \in (0,1)$ for $s > 0$, being $s$ the number of sub-steps used to compute the given flux. However, if $\Delta t < \Delta t^*$ all fluxes will be accepted straightaway,

so $s = 0$ and $\alpha_0 = 1$. That is, for $\Delta t < \Delta t^*$ the scheme recovers the basic scheme, which is fully consistent. □

We mention that the condition on the error indicators $\epsilon_{i+1/2}$ is also necessary to ensure that our multiscale algorithm terminates.

### 2.4.3 Von Neumann stability analysis

In this section we will analyze the stability of the conservative multirate approach using, as method to integrate in time, the Implict Euler, and a fixed number of sub-refinements. We consider the model problem

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \qquad x \in \mathbb{R},\ t > 0$$

To discretize in space a first order upwind scheme has been used.

**Proposition 2.4.5.** *Assuming that the initial condition is 2-$\pi$ periodic so that can be expanded in a Fourier series:*

$$u_0(x) = \sum_{k=-\infty}^{\infty} \alpha_k e^{ikx}$$

*where*

$$\alpha_k = \frac{1}{2\pi}\int_0^{2\pi} u_0(x)e^{-ikx}\mathrm{d}x$$

*is the $k$-th Fourier coefficient of $u_0$. A numerical scheme is stable with respect to the $||\cdot||_{\Delta,2}$ norm, if the absolute value of the amplification coefficient $|\gamma_k| \leq 1\ \forall k$, where $||\cdot||_{\Delta,2}$ is the following discrete norm:*

$$||v||_{\Delta,2} = \left(\Delta x \sum_{j=-\infty}^{\infty} |v_j|^p\right)^{\frac{1}{p}}.$$

For detail see [81].

**Definition 2.4.2.** *The* amplification coefficient *of the $k$-th frequency (or harmonic) $\gamma_k \in \mathbb{C}$ is defined as the ratio between the $k$-frequency of the current time step $[u_i^{n+1}]$ and the $k$-th frequency of the previous step $[u_i^n]$:*

$$\gamma_k = \frac{[u_i^{n+1}]_k}{[u_i^n]_k}.$$

Considering a 1-step of sub-refinement of the multirate method, a global time step for the linear advection equation ($a > 0$) reads as:

$$u_i^{n+1} = u_i^n - \left\{F_{i+\frac{1}{2}}^{n+1} - F_{i-\frac{1}{2}}^{n+1}\right\} = u_i^n - \left\{\lambda a\left(u_i^{n+1} + u_{i-1}^{n+1}\right)\right\}$$

where $\lambda = \dfrac{\Delta x}{\Delta t}$. Let us suppose, as for the consistency analysis, that the numerical flux $F_{i-\frac{1}{2}}^{n+1}$ is accepted after the tentative global time step and,

instead, the flux $F_{i+\frac{1}{2}}^{n+1}$ needs to be recomputed; to write the amplicitation coefficient, first of all we need the $k$-th frequency $[u_{i-1}^{n+1}]_k$, written respect to the $k$-th frequency $[u_i^n]_k$:

$$\begin{aligned}
[u_{i-1}^{n+1}]_k &= [u_{i-1}^n]_k - a\lambda\Big\{[u_{i-1}^{n+1}]_k - [u_{i-2}^{n+1}]_k\Big\} \\
&= [u_i^n]_k e^{-ik\Delta x} - a\lambda\Big\{[u_{i-1}^{n+1}]_k - [u_{i-1}^n]_k e^{-ik\Delta x}\Big\} \\
&= \frac{e^{-ik\Delta x}}{(1+a\lambda\,(1-e^{-ik\Delta x}))}[u_i^n]_k
\end{aligned} \tag{2.11}$$

Computing the first sub-refinement step with a size step equal to $\frac{\Delta t}{2}$, the $k$-th frequency for the solution $[u_i^{n+\frac{1}{2}}]_k$ as function of $[u_i^n]_k$ would be:

$$[u_i^{n+\frac{1}{2}}]_k = [u_i^n]_k - a\frac{\lambda}{2}\Big\{[u_i^{n+\frac{1}{2}}]_k - [u_{i-1}^{n+1}]_k\Big\},$$

using the expression (2.11) we obtain:

$$[u_i^{n+\frac{1}{2}}]_k = \left(1+\frac{\lambda}{2}\right)^{-1}\left\{1+a\frac{\lambda}{2}\frac{e^{-ik\Delta x}}{1+a\lambda\,(1-e^{-ik\Delta x})}\right\}[u_i^n]_k. \tag{2.12}$$

If the numerical flux $F_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ would be accepted, the final $k$-th frequency reads:

$$[u_i^{n+1}]_k = 4[u_i^{n+\frac{1}{2}}]_k - a\frac{\lambda}{2}\Big\{[u_i^{n+1}]_k - [u_{i-1}^{n+1}]_k\Big\}$$

using expressions (2.12) and (2.11) we finally obtain:

$$\begin{aligned}
[u_i^{n+1}]_k = \Bigg\{ &\left(1+\frac{\lambda}{2}\right)^{-2}\left\{1+a\frac{\lambda}{2}\frac{e^{-ik\Delta x}}{1+a\lambda\,(1-e^{-ik\Delta x})}\right\} \\
&+ a\frac{\lambda}{2}\left(1+\frac{\lambda}{2}\right)^{-1}\frac{e^{-ik\Delta x}}{1+a\lambda\,(1e^{-ik\Delta x})}\Bigg\}[u_i^n]_k
\end{aligned}$$

Generalizing the process with $m$ sub-refinements with fixed internal step size equal to $\frac{\Delta t}{m}$ the expression for the amplification coefficient will be:

$$\begin{aligned}
\gamma_k = &\left(1+\frac{\lambda}{m}\right)^{-m}\left\{1+a\frac{\lambda}{m}\frac{e^{-ik\Delta x}}{1+a\lambda\,(1-e^{-ik\Delta x})}\right\} \\
&+ \sum_{s=1}^{m-1} a\frac{\lambda}{m}\left(1+\frac{\lambda}{m}\right)^{-s}\frac{e^{-ik\Delta x}}{1+a\lambda\,(1-e^{-ik\Delta x})}.
\end{aligned}$$

As said before, the absolute value of the modulus of a numerical method has to be lower or equal to one to be the method stable and, moreover, if it satisfies the condition for any choice of $\Delta t$ and $\Delta x$ it is unconditionally stable.

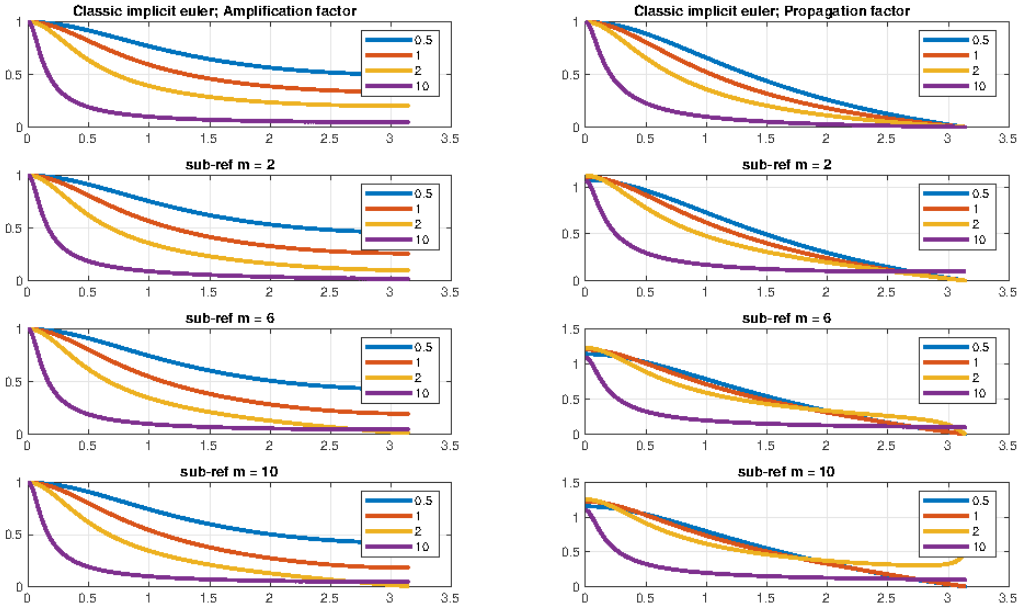


**Fig. 2.3.:** Amplification coefficients (left column) and propagation factor (right column), for different number of sub-refinements and for different choices of $a\lambda$.

It is difficult to compute when $|\gamma_k| \leq 1$ analytically so, in Fig. 2.3, we plotted the amplification errors and the dispersion errors for different values of $a\lambda$. The *amplification error* in our case coincides with the amplification coefficient, as explained in [81], instead the *dispersion coefficient* is the ratio between the velocity of propagation of the numerical solution relative to its $k$-th harmonic $\frac{\omega}{k}$ and the velocity $a$ of the exact solution.

The amplification and dispersion errors are functions of the phase angle $\varphi_k = k\Delta x$ that varies on the interval $0 \leq \varphi_k \leq \pi$. We can see that, even if we increase the number of sub-refinements, for all Courant number $a\lambda$ the amplification coefficient is below one.

Fig. 2.4 shows the exact solution (dashed line) and the numerical solutions (solid line) at $t = 1$ for the problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial u}{\partial x} = 0 & x \in [-1, 3], t > 0 \\ u(x, 0) = \begin{cases} \sin(2\pi x) & x \in [-1, 1] \\ 0 & \text{otherwise.} \end{cases} \end{cases}$$

For each column the CFL number is fixed and the number of sub-refinements is varying. First row shows the results for the backward Euler approach without sub-refinements. The solution is more dissipate as long as the CFL value has been increased. This shows how the use of multirate technique allows to reduce the overall numerical dissipation of the basic scheme.

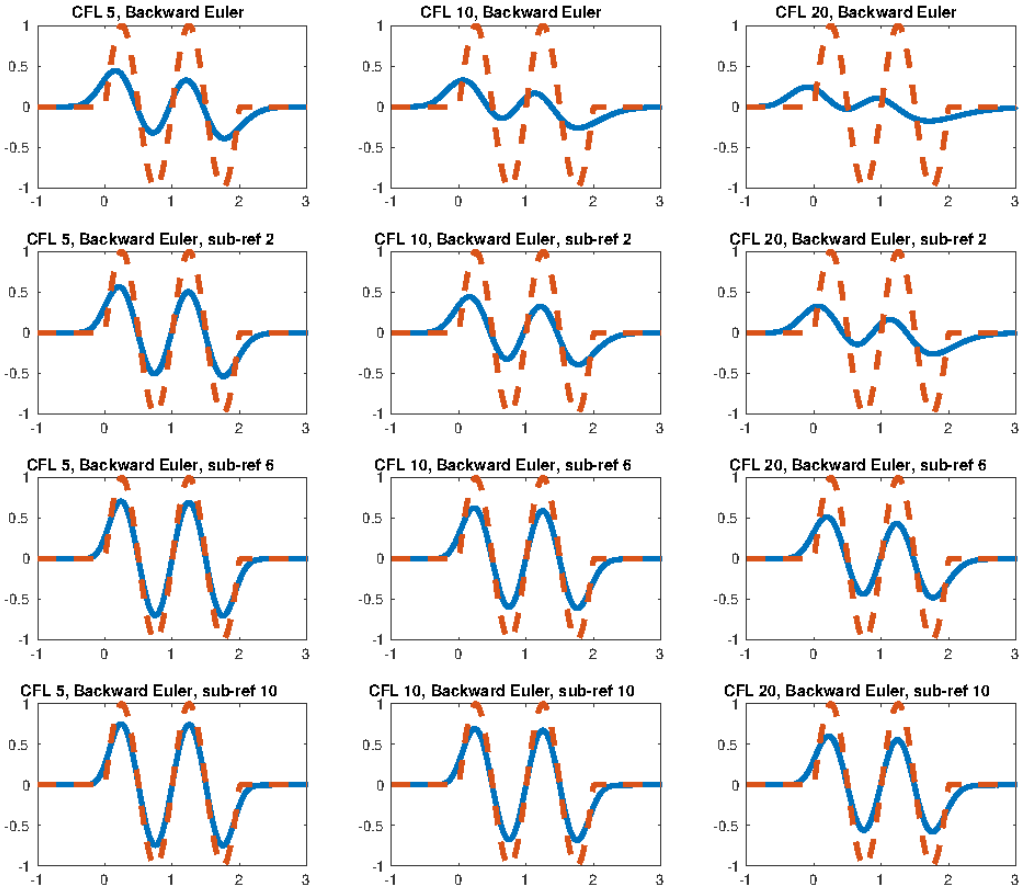


**Fig. 2.4.:** Exact solution (dashed line) and numerical solution (solid line) at $t = 1$ for the advection problem with $a = 1$ for different global CFL values.

## 2.5 Time discretization with TR-BDF2

While a time discretization based on the $\theta-$method has been employed to introduce the proposed conservative multirate method and for its analysis in term of consistency and stability, in this chapter we focus on the use of multirate TR-BDF2 method, as in [9], because of its interesting properties. This method is a composite one step, two stages method, consisting of one stage of the trapezoidal scheme followed by one stage of the BDF2 method. It can be written for the discretization of an ODE system $y' = f(t, y)$ as

$$u^{n+\gamma} = u^n + \frac{\Delta t_n \gamma}{2}\left(f(t^n, u^n) + f(t^{n+\gamma}, u^{n+\gamma})\right)$$
$$u^{n+1} = \frac{1}{\gamma(2-\gamma)}u^{n+\gamma} - \frac{(1-\gamma)^2}{\gamma(2-\gamma)}u^n + \frac{1-\gamma}{2-\gamma}\Delta t_n f(t^{n+1}, u^{n+1})$$

For $\gamma = 2 - \sqrt{2}$, the method is L-stable and also employs the same Jacobian matrix for the two stages. In [50] it has been interpreted as a Diagonally Implicit Runge Kutta (DIRK) method with two internal stages, proving the following properties:

- the method is strongly S-Stable;
- it is endowed with a Cubic Hermite interpolation algorithm that yields globally $\mathcal{C}^1$ continuous trajectories.

Due to its favorable properties, it has been recently applied for efficient discretization of high order finite element methods for numerical weather forecasting in [93], while its monotonicity properties have been studied in [8].

### 2.5.1 Flux-partitioning and error estimator

To select the components that have to be recomputed with a smaller time step, we need to introduce a local error estimator for the fluxes. A simple approach is to compare the fluxes computed with the $\theta$-method or the TR-BDF2 method, with the fluxes at the same interface cell computed with a more accurate method. The absolute value of the difference between the two fluxes can be used as a measure of the error. For $\gamma = 2 - \sqrt{2}$ the TR-BDF2 scheme has a third order method embedded, this fact can be exploited to derive the error estimator, yet as remarked in [50], the third order method embedded in TR-BDF2 is not A-stable. In that work a heuristic approach that entails the solution of an additional linear system per time step has been proposed to stabilize the error estimator. For a large ODE systems coming from the spatial discretization of PDEs, solving at each time step this extra linear system could turn out to be very expensive.

Therefore, we propose other types of error estimator, which are less expensive. At each time step, for a two stage method as the TR-BDF2 method, we know the active components values at times $t^n$ and $t^{n+\gamma}$ , so we can use an extrapolation technique to obtain a prediction of the value at time $t^{n+1}$. If

we call the extrapolated solution at time $t^{n+1}$ as $\bar{u}_{ext}^{n+1}$, the extrapolated fluxes at the interface are $\bar{F}_{ext_{i+\frac{1}{2}}}^{n+1}$ and we obtain the error estimator as:

$$\mathcal{R}_F = \{F_{i+\frac{1}{2}} : |F_{i+\frac{1}{2}}^{n+1} - \bar{F}_{ext_{i+\frac{1}{2}}}^{n+1}| > \tau_r |F_{i+\frac{1}{2}}^{n+1}| + \tau_a\}$$

where $\tau_r > 0$ is a relative tolerance, and $\tau_a > 0$ is needed to handle the case $|F_{i+\frac{1}{2}}^{n+1}| = 0$.

The simplest extrapolation technique is the linear extrapolation, given by

$$\bar{u}_{lin}^{n+1} = u^n + \frac{t^{n+1} - t^n}{t^{n+\gamma} - t^n} \left(u^{n+\gamma} - u^n\right),$$

by which we obtain the extrapolated values of $F_{lin_{i+\frac{1}{2}}}^{n+1}$ at the required interface, whose difference with the computed value provides the error estimator.

A more precise estimator can be obtained by applying a cubic Hermite extrapolation at time $t^n$ and $t^{n+\gamma}$ considering the fact that the TR-BDF2 method provides a formula to compute the coefficient for the cubic Hermite extrapolation easily.

The extrapolation can be evaluated as:

$$\bar{u}_{cub}(t) \quad = \quad (\alpha_3 - 2\alpha_2)\beta(t)^3 + (3\alpha_2 - \alpha_3)\beta(t)^2 + \alpha_1\beta(t) + \alpha_0,$$

$\alpha$ coefficients are:

$$\alpha_0 = u^n, \quad \alpha_1 = \gamma \Delta t_n f(t_n, u^n), \quad \alpha_2 = u^{n+\gamma} - u^n - \alpha_1,$$
$$\alpha_3 = \gamma \Delta t_n (f(t^{n+\gamma}, u^{n+\gamma}) - f(t^n, u^n)),$$

instead $\beta$ is:

$$\beta(t) = \frac{t - t^n}{\gamma \Delta t_n}.$$

At time $t^{n+1}$ the extrapolated solution would be:

$$\bar{u}_{cub}^{t+1} = (\alpha_3 - 2\alpha_2)\left(\frac{1}{\gamma}\right)^3 + (3\alpha_2 - \alpha_3)\left(\frac{1}{\gamma}\right)^2 + \alpha_1\left(\frac{1}{\gamma}\right) + \alpha_0,$$

In our test cases we use the error estimator based on the Cubic Hermite extrapolation.

### 2.5.2 Systems of PDEs

The multirate method is easily extended to a system of nonlinear conservation laws. The only non trivial part is how to define the set of active fluxes.

A system of $d$ nonlinear conservation laws can be written as:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial(\mathbf{f}(\mathbf{u}))}{\partial x} = 0 \qquad x \in \mathbb{R} \quad t > 0 \tag{2.13}$$

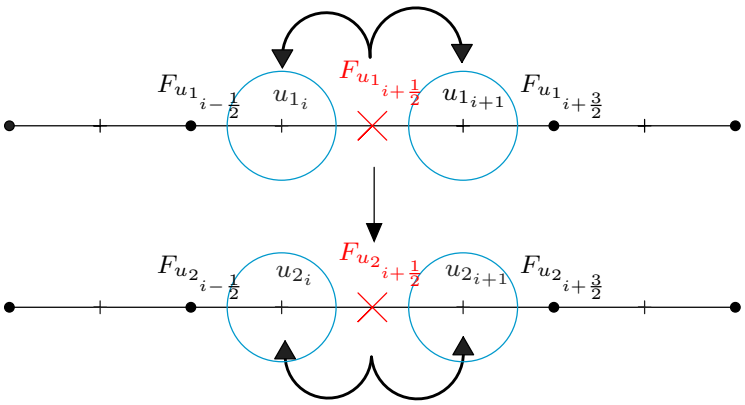


**Fig. 2.5.:** Example of rejected fluxes in a system of non-linear conservation laws.

where $\mathbf{u}$ and $\mathbf{f}$ are $d$-vectors on the problem domain, $\mathbf{u} = [u_1, u_2, \cdots, u_d]^T$ and

$$\mathbf{F}(\mathbf{u}) = [F_1(u_1, \cdots, u_d), F_2(u_1, \cdots, u_d), \cdots, F_d(u_1, \cdots, u_d)]^T$$

is a vector of fluxes.

If we use a two-point flux approximation, when (2.13) is semi-discretized in space, the numerical flux at each interface $x_{i+\frac{1}{2}}$ depends on the approximation $\mathbf{u}_i$ and $\mathbf{u}_{i+1}$ at nodes $x_i$ and $x_{i+1}$, respectively. To preserve the mass of the whole system, if the $j$-th components of the flux for the $i$-th variable has been rejected, all components should be considered as rejected.

In Fig. 2.5, we show a simple example with $d = 2$. If the flux for the variable $u_1$ has been rejected in position $x_{i+\frac{1}{2}}$, the components $u_{1_i}$ and $u_{1_{i+1}}$ will be included in the set of active components but, to be conservative, also the flux for the variable $u_2$ will be rejected and so also the components $u_{2_i}$ and $u_{2_{i+1}}$ will be recomputed with a smaller time step.

### 2.5.3 Boundary conditions

To illustrate our scheme we have assumed that the differential problem is set on the whole real line. However, in the numerical tests of the next Section (as well as in all practical situations) we have to deal with bounded domain, and proper boundary conditions must be imposed. Since we are adopting a finite volume scheme, the boundary conditions have been applied by computing the fluxes at the fictitious boundary interface by the well known "ghost node" technique. With this method the correct type of information (i.e. that corresponding to the characteristics entering the domain) is automatically selected by the numerical scheme.

It may happen that a flux between a ghost node and a real node is rejected inside a generic time step. In this case, the ghost cell is evaluated at the internal time and the flux is computed with a smaller time step as an internal flux of the domain.

## 2.6 Numerical results

In this section, we present different numerical experiments to test the efficiency and the accuracy of the conservative multirate method. First we apply it to the Burgers' equation, then to a more complex scalar test case, the Buckley-Leverett equation and, at the end, we illustrate the multirate method applied to a system of nonlinear conservation laws, the Shallow Water equations.

### 2.6.1 Test case 1: Burgers equation

Here, we apply the multirate method to Burgers equation with Dirichlet boundary conditions, thus repeating the tests presented in [9], but with the conservative variant of our algorithm. The Burgers equation is a nonlinear conservation law and we consider the following setting:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{1}{2}u^2\right) = 0 & (x,t) \in (-1,3)\times(0,1), \\ u(x,0) = u_0(x) & x \in (-1,3), \\ u(-1,t) = u_l(t) \quad u(3,t) = u_r(t) & t \in (0,1), \end{cases}$$

where $u^0(x) = \begin{cases} u_l(0) & x < 0, \\ u_r(0) & x > 0. \end{cases}$

#### First case: $u_l > u_r$

In this case we consider $u_l = 1$ and $u_r = 0$ with a number of cells equal to $400$, the absolute and relative error tolerances are $\tau_a = 10^{-4}$, $\tau_r = 10^{-6}$, respectively, while the tolerance for the Newton solver is $10^{-14}$ on the difference between two consecutive iterations. The TR-BDF2 method has been used as solver to integrate in time, the size of the global time step is equal to $0.1$. To obtain an entropic solution we used the local Lax Friedrichs flux [91] (also know as Rusanov flux) as numerical flux for the two point Finite Volume method:

$$F_{i+\frac{1}{2}} = F_{i+\frac{1}{2}}(u_i, u_{i+1}) = \frac{1}{2}\left[(f(u_{i+1}) + f(u_i)) - \alpha(u_{i+1} - u_i)\right], \tag{2.14}$$

where $\alpha = \max_{\omega} |f'(\omega)|$ and the maximum is taken in the range $\omega \in [u_i, u_{i+i}]$. As we can see in Fig. 2.6, the solution computed with the multirate method is in excellent agreement with the exact solution. In Fig. 2.7 we represent the set of active components at each time. We can observe that the multirate method captures the shock and refines only the region of the domain where the solution is changing rapidly. We also plot the Courant numbers for each time step, Fig. 2.7. The self-adjusting strategy selects small Courant numbers inside the time slab in the locations near the discontinuity, while the global step corresponds to a Courant number $2.5$. Note that we prescribed a global step size equal to $0.1$, that gives a Courant number of $10$, but all components have been rejected for the given value of the error tolerance, so that the

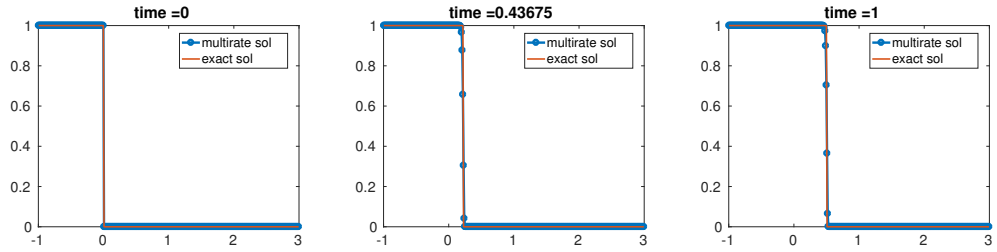


**Fig. 2.6.:** Test case 1 $u_l > u_r$ – Multirate TR-BDF2 integration and exact solution for the shock wave at different times $t = 0$s, $0.45$s and $1$s.

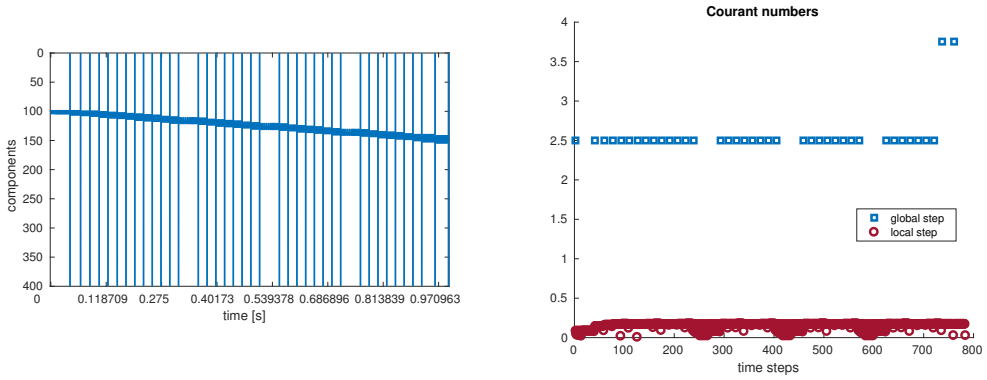


**Fig. 2.7.:** Test case 1 $u_l > u_r$ – The components being computed at each time step by the TR-BDF2 method (left) and Courant number for each time step (right).

global time step size is in fact smaller and equal to $0.025$ except for the last two time slabs.

## Second case: $u_r > u_l$

We set the value at the left $u_l = 0$ and the value at the right $u_r = 1$. In this case, the solution exhibits a rarefaction wave. The boundary conditions are $u(-1, t) = u_l \quad \forall t \in (0, 1)$ and $u(3, t) = u_r \quad \forall t \in (0, 1)$, while the other parameters are the same as in the previous test case. In Fig. 2.8 we can see the solution obtained with the multirate method. Some numerical diffusion is clearly visible due to the first order monotone flux employed. In this case, the Courant number for the global step is equal to $10$, as shown in Fig. 2.9. The Courant numbers for the sub-steps inside the time slab are larger than those obtained in the shock wave solution and consequently less time steps are necessary to compute the solution at the final time. Fig. 2.9 also represents the set of active components at each time. As expected, the size of the set increases with time because the rarefaction zone is expanding.

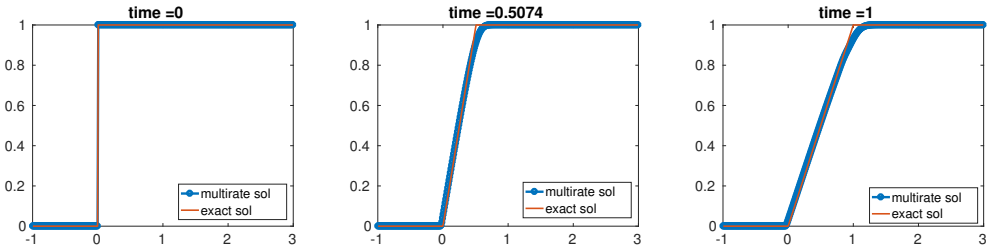


**Fig. 2.8.:** Test case 1 $u_r > u_l$ – Multirate TR-BDF2 integration and the exact solution for the rarefaction wave at different times.

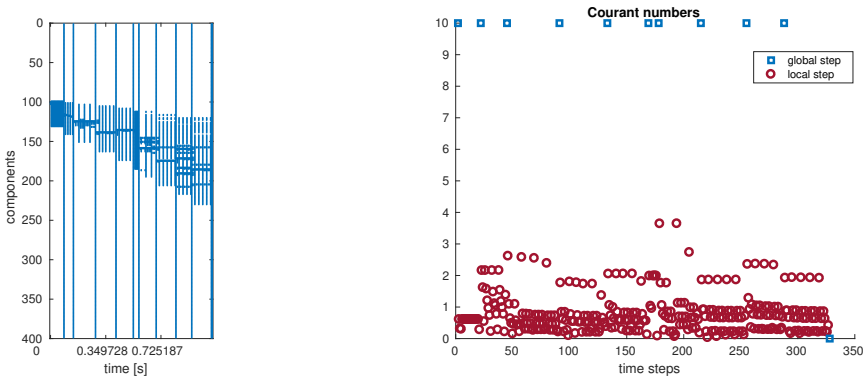


**Fig. 2.9.:** Test case 2 $u_r > u_l$ – The components being computed at each time step with the TR-BDF2 method (left) and Courant number for each time step (right).

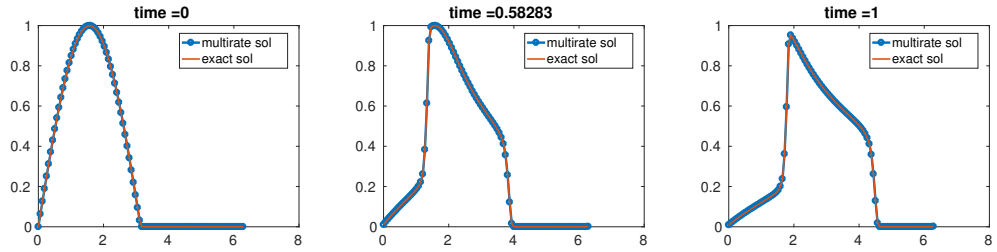


**Fig. 2.10.:** Test case 2 – Multirate TR-BDF2 solution and the solution computed with the `ode45` matlab solver.

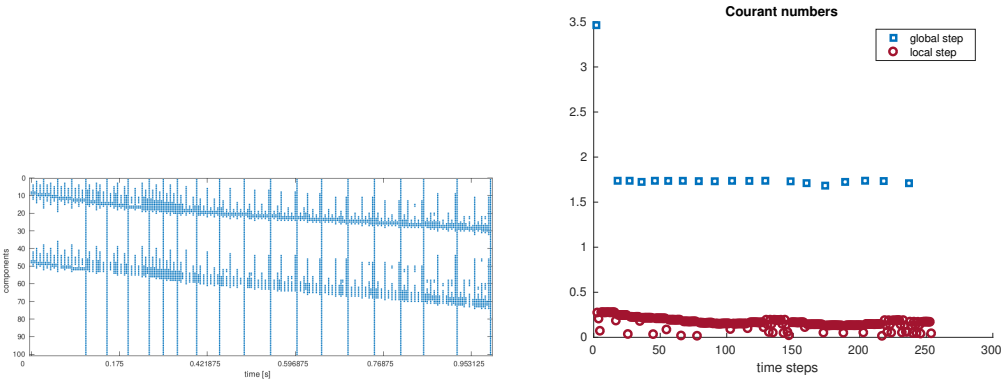


**Fig. 2.11.:** Test case 2 – The components being computed at each time step (left) and Courant number for each time step (right).

## 2.6.2 Test case 2: Buckley-Leverett equation

An example of a more complex conservation law is given by the Buckley-Leverett equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial}{\partial x} f(u) = 0 & (x,t) \in (0, 2\pi) \times (0,1) \\ f(u) = \dfrac{u^2}{u^2 + \frac{1}{3}(1-u)^2} & \\ u(x,0) = \sin(x) & x \in (0, 2\pi) \\ u(0,t) = u(2\pi, t) & t \in (0,1). \end{cases}$$

Also in this case, we use a two-point finite volume scheme with Rusanov flux, and $N_x = 100$ cells. The TR-BDF2 method is used to integrate up to time $T = 0.5$ with a global size step $\Delta t = 0.1$. In this case, periodic boundary conditions are employed. The absolute and relative error tolerances are $10^{-4}$, $10^{-5}$, respectively, while the tolerance for the Newton solver is $10^{-13}$. To compute the l1-norm of the error we use as a reference solution provided by the Matlab solver `ode45` with maximum allowed time step equal to $\Delta t = 10^{-5}$.

This is a rather complex test case, because of the presence of both a shock and a rarefaction wave, as we can see in Fig. 2.10. The multirate method refines the time-steps only where the solution is fast moving, as indicated by the smaller Courant numbers Fig. 2.11.

**Tab. 2.1.:** Test case 2 – The ratio between the initial and final mass, the normalized difference between the initial and final mass in the Buckley-Leverett equation test case.

| | ratio | diff. | $l1$-norm |
|---|---|---|---|
| MC scheme | 1 | $8.36e-15$ | 0.0013 |
| N-MC scheme | 0.96 | 0.0313 | 0.0012 |

We then compare our mass conservative approach with the original multirate method proposed in [9]. As shown in Tab. 2.1, we obtain essentially the same error in the $l_1$-norm for both methods, but, while with the previous method the system loses $4\%$ of the mass during the simulation, the new method preserves the total mass of the system, as expected.

### 2.6.3 Test case 3: Saint-Venant equations or dam break problem

We consider now our multirate strategy applied to the Saint-Venant (or shallow water) equations, which can be written in conservative form as:

$$\begin{cases} \dfrac{\partial h}{\partial t} + \dfrac{\partial q}{\partial x} = 0 \\ \dfrac{\partial q}{\partial t} + \dfrac{\partial}{\partial x}\left(\dfrac{q^2}{h} + g\dfrac{h^2}{2}\right) = 0. \end{cases}$$

Here, $h$ denotes the fluid depth and $q = hu$ the discharge, where $u$ is the velocity of the fluid. These equations are the core of many numerical models for river hydraulics and environmental flows. A more complete discussion of the Saint-Venant equations can be found in [64]. It has to be remarked that even very efficient single rate semi-implicit methods, see e.g. [84], when applied to the Saint-Venant equations in presence of shocks, must employ small time steps throughout the domain to achieve a reasonable accuracy. As we will see, this shortcoming is overcome by our approach.

The dam break problem is a special case of the Riemann problem, where at the initial time $h^0(x) = \begin{cases} h_l & \text{if } x < x_0 \\ h_r & \text{if } x > x_0 \end{cases}$ and $u = q = 0$ everywhere in the domain. For the spatial discretization of the Saint-Venant equations we used again the Rusanov flux. In this case, the numerical diffusion coefficient $\alpha$ in (2.14) is defined as:

$$\alpha = max\{|\lambda_i^1|, |\lambda_i^2|, |\lambda_{i+1}^1|, |\lambda_{i+1}^2|\},$$

$\lambda_i^1$ and $\lambda_i^2$ are eigenvalues of the system for the control volume $i$:

$$\lambda_i^1 = \frac{h_i}{q_i} - \sqrt{g h_i}$$
$$\lambda_i^2 = \frac{h_i}{q_i} + \sqrt{g h_i}.$$

We used 300 cells over the domain $(0, 3000)$ [m], while the absolute and relative error tolerances are $\tau_a = 10^{-2}$, $\tau_r = 10^{-4}$, respectively, while the tolerance for the Newton solver is $10^{-13}$. The size of the global steps is equal to $8$s, and we integrate in the time interval $(0, 100]$ [s]. The initial condition for the water height is $h^0(x) = \begin{cases} 1.5 & \text{if } x < 1500 \\ 0 & \text{if } x > 1500 \end{cases}$ [m], and for the water velocity we set $u = \frac{q}{h} = 0$ [m/s].

When performing this test with the original version of the algorithm described in the previous sections, numerical oscillation across the boundary between the refinement and the non-refinement regions were observed. These oscillations are due to the fact that the error estimator accepted some fluxes that were changing their values inside the time slab and it was not correct to use their final time slab values for the entire considered sub-step. To avoid this problem, we slightly modified the set of rejected fluxes. If a flux is rejected, we also reject a number of fluxes (on the left or on the right or on both sides, depending on the sign of the eigenvalues) equal to the local Courant number. In this way, as shown in Fig. 2.12, the solution has the correct behavior; of course, we are increasing the set of active components, but the latent components are still the majority during time integration (Fig. 2.13). It can be seen clearly that, as in the scalar case, the method is able to identify the complex nonlinear features of the flow automatically. It can also be seen in Fig. 2.13 that a Courant number larger than one was allowed for the global time steps without any significant loss in accuracy.

### 2.6.4 Test case 4: Shallow water equations with rotation

We have also considered the shallow water equations with rotation, which are a classical idealized model for the phenomenon of geostrophic adjustment, see e.g. [41]. This system, in the semi-linear form obtained discarding the nonlinear momentum advection terms, can be written as:

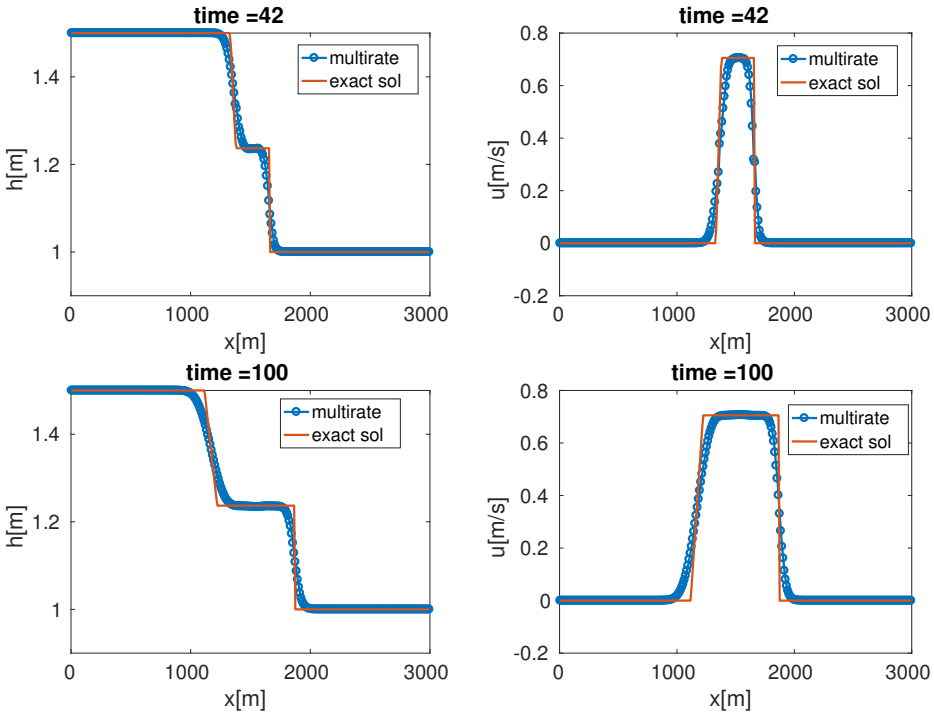


**Fig. 2.12.:** Test case 3 – Solutions at time $t = 42$ [s] and $t = 100$ [s] for the $h$ variable (on the left) and for $u = \frac{q}{h}$ variable (on the right).

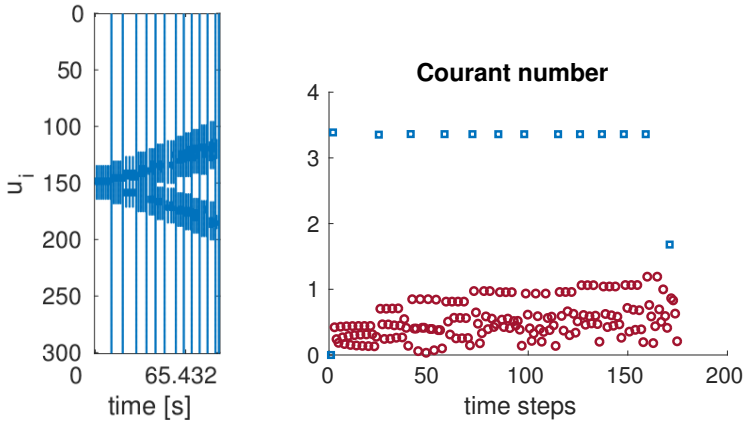


**Fig. 2.13.:** Test case 3 – Set of active components for one variable at each time (left) and Courant number for each time step (right).

$$
\begin{cases}
\dfrac{\partial \eta}{\partial t} + \dfrac{\partial((\eta+\eta_0)u)}{\partial x} = 0 & (x,t) \in (-L,L)\times(0,T) \\
\dfrac{\partial u}{\partial t} + g\dfrac{\partial \eta}{\partial x} + fv = 0 & (x,t) \in (-L,L)\times(0,T) \\
\dfrac{\partial v}{\partial t} - fu = 0 & (x,t) \in (-L,L)\times(0,T) \\
\eta(x,t=0) = \exp\left(-\dfrac{(50x)^2}{(2L)^2}\right) & x \in (-L,L) \\
u(x,t=0) = v(x,t=0) = 0 & x \in (-L,L) \\
\eta(-L,t) = \eta(L,t) = 0 & t \in (0,T) \\
u(-L,t) = u(L,t) = 0 & t \in (0,T) \\
v(-L,t) = v(L,t) = 0 & t \in (0,T)
\end{cases}
$$

Here, $\eta$ denotes the free surface height, $u$ the velocity in the $x$ direction, $g$ the gravity acceleration, $f$ a constant Coriolis parameter and $v$ the velocity in the direction orthogonal to the one dimensional flow being considered. This system is of particular interest since it describes a dynamics with two different time scales, a fast one associated to the propagation of external gravity waves and a slow one associated with rotational effects and the onset of geostrophic equilibrium. Semi-implicit techniques commonly applied for geophysical scale flows (see e.g. the classical paper [42]) allow to achieve an accurate approximation of the slow components, while sacrificing the accuracy of the fast ones.

In order to represent a large geophysical scale, we have used $L = 8 \times 10^6$ [m] , $T = 3 \times 10^6$ [s], $f = 10^{-4}$ 1 [s] and $\eta_0 = 1000$ [m]. We have discretized in space with $N_x = 480$ cells and we have used, as space discretization, the following conservative centered finite difference scheme:

$$
\begin{aligned}
\frac{d\eta_i}{dt} &= -\left[\frac{u_i\eta_i + u_{i+1}\eta_{i+1}}{2\Delta x} - \frac{u_i\eta_i + u_{i-1}\eta_{i-1}}{2\Delta x}\right], \\
\frac{du_i}{dt} &= -g\left[\frac{\eta_i + \eta_{i+1}}{2\Delta x} - \frac{\eta_i + \eta_{i+1}}{2\Delta x}\right] - fv_i, \\
\frac{dv_i}{dt} &= fu_i.
\end{aligned}
$$

In this case, we used a global step $\Delta t = 700$ [s] to discretize in time. The solution is represented in Fig. 2.14, while the set of active/refined components for the $\eta$ variable is displayed in Fig. 2.15. It can be seen that, also in this case, the proposed algorithm is able to identify automatically the different time scales present in the solution. The component of the solution at the center of the domain, which tends to geostrophic equilibrium on a slow time scale, does not require any refinement of the time step, while the fast propagating gravity waves induce refinement along the wave trails. Notice that we plot the active components for the $\eta$ variable only because the set of active components and active fluxes are the same for each variable of the system in order to preserve mass. It can also be seen in Fig. 2.15 that Courant

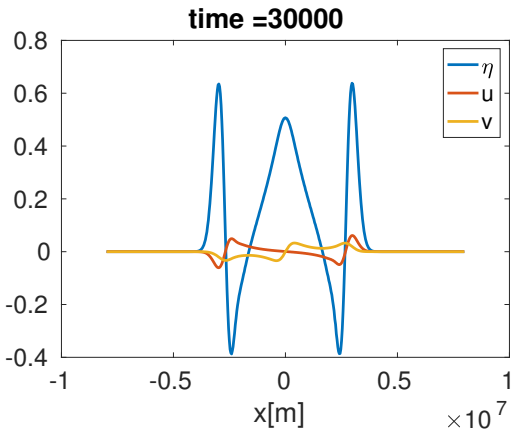


**Fig. 2.14.:** Test case 4 – Solutions at the final time computed with the multirate method.

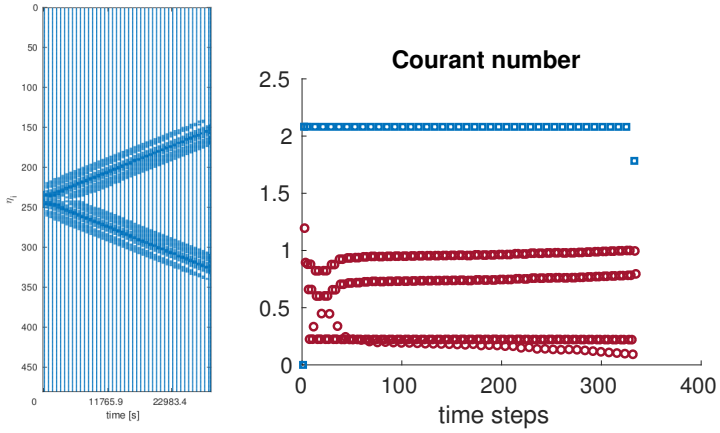


**Fig. 2.15.:** Test case 4 – Set of active components for $\eta$ variable (left) and Courant number for each time step (right).

numbers larger than one are feasible for the global time steps without any significant loss in accuracy.

In Tab. 2.2 we reported the comparison with the single-rate version of the TR-BDF2 method. In the first column we report the CPU time required to solve the problem with the two different methods. It has to be remarked that both methods were implemented in a rather straightforward way and that the respective codes are far from optimized. On the other hand, exactly the same computational components, such as e.g. the Newton solver, were employed in both, so that the ratio of the CPU times required by the two approaches is a reasonable estimate of the potential speed-up. It can be seen that the multirate approach solves the problem more than twice as fast as the single rate method.

The multirate method refines the time steps only for a small portion of components of the solution, while allowing a larger time step for the other components. Indeed in Tab 2.2 we may note that the total number of function evaluation is smaller for the multirate computation, and this reflects on the computational time.

**Tab. 2.2.:** Test case 4 – Computational time and total components number involved, using a relative tolerance equal to $\tau_r = 10^{-4}$ and as absolute tolerance $\tau_a = 10^{-3}$ for both the single rate and the multirate approach.

| | comp. time [s] | # function eval. |
|---|---|---|
| Multirate | 74.71 | 102336 |
| Single rate | 179.29 | 186810 |

# 3 Modeling fluid flow in porous media and classical discretization techniques

In this chapter the governing equations used to model flow and transport processes in porous media are provided. The classical numerical strategies to solve the problem are discussed and, finally, the discretization techniques used in this thesis are presented.

## 3.1 Single phase flow

Single phase flow in porous media is described by the mass conservation equation:

$$\frac{\partial}{\partial t}(\phi\rho) + \nabla \cdot (\rho \mathbf{u}) = \rho q \qquad \text{in } \Omega,\ t > 0,$$

where $\phi$ is the porosity, $\rho$ is the density and $q$ the source term.

The velocity is expressed by the Darcy velocity:

$$\mathbf{u} = -\frac{1}{\mu}\mathbf{K}\left(\nabla p - \rho g \nabla z\right) \qquad \text{in } \Omega \subset \mathbb{R}^3,$$

where $\mu$ is the viscosity. The rock permeability $\mathbf{K}$ is a heterogeneous symmetric-positive definite tensor. It means that there are two positive constant $k_{min}$ and $k_{max}$ such that $k_{min} \leq v^T \mathbf{K}(x) v \leq k_{max}$ for $x$ almost everywhere in $\Omega$ and for any $v \in \mathbb{R}^3$. Finally, $g$ is the gravitational constant and $z$ a vector pointing in the direction of the gravitational force. The problem is completed by appropriate initial and boundary conditions.

## 3.2 Multiphase flow

Let $N_p$ be the number of phases present in the fluid, which we assume is filling the whole pore space. The mass balance for immiscible fluids now reads

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha \mathbf{u}_\alpha) = \rho_\alpha q_\alpha, \qquad \alpha \in \{1, ..., N_p\},$$

where $S_\alpha$ is the saturation, $\mathbf{u}_\alpha$ the Darcy velocity, $\rho_\alpha$ the density and $q_\alpha$ the source term for phase $\alpha$.

The Darcy phase velocity is given by

$$\mathbf{u}_\alpha = -\lambda_\alpha \mathbf{K}\left(\nabla p_\alpha - \rho_\alpha g \nabla z\right) \qquad \text{in } \Omega,$$

where $\lambda_\alpha = k_{r\alpha}/\mu_\alpha > 0$ is the phase mobility – i.e. the ratio of the relative permeability $k_{r\alpha}$ (which is a function of the phase saturations [12]) – and the phase viscosity $\mu_\alpha$. Phase pressures $p_\alpha$ are related by capillary pressure $P_c$:

$$p_\alpha - p_\beta = (1 - \delta_{\alpha,\beta}) P_{c_{\alpha,\beta}} \qquad \forall \alpha, \beta \in 1, \dots, N_p. \tag{3.1}$$

$\delta_{\alpha,\beta}$ is the Kronecker delta, equal 1 if $\alpha = \beta$ and 0 otherwise. $P_{c_{\alpha,\beta}}$ is a nonlinear function of the wetting saturation. The saturation of the phases must fulfill the constraint

$$\sum_{i=1}^{N_p} S_i = 1 \tag{3.2}$$

i.e. the pore space is fully filled with fluid phases. The set of equations represent a coupled system of differential equations that is strongly nonlinear because of the nonlinear dependence on the saturation of the capillary pressures and the relative permeabilities.

These equations together with proper boundary and initial conditions, form a well-posed coupled system for $N_p$ unknowns. Let the boundary $\partial\Omega$ of the computational domain $\Omega$ be decomposed into a non-overlapping Dirichlet $\Gamma_D$ and Neumann $\Gamma_N$ parts, where $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. Possible boundary conditions for the equations read, for $t > 0$

$$\begin{cases} p_\alpha = p_{\alpha_D} \text{ and } S_\alpha = \bar{S}_\alpha & \text{on } \Gamma_D, \\ \rho_\alpha \mathbf{u}_\alpha \cdot \mathbf{n} = h_\alpha & \text{on } \Gamma_N. \end{cases} \tag{3.3}$$

The initial conditions at time $t = 0$ are

$$S_\alpha = S_\alpha^0 \text{ and } p_\alpha = p_\alpha^0 \qquad \text{in } \Omega,$$

The boundary conditions (3.3) must be compatible with the algebraic constraints (3.1) and (3.2). Only $N_p$ variables can be chosen as independent unknowns. In literature are present different formulations depending on the choice of the independent variables or primary unknowns. For an introduction to different formulations see [14].

### 3.2.1 Capillary pressure curves

If we consider a generic two-phase system with a wetting (w) and a non-wetting (nw) phase, we need a single capillary pressure curve $P_c = p_{nw} - p_w$. In general, capillary pressure may depend on temperature and fluid composition, but in this work we consider only a dependence on wetting saturation $P_c = P_c(S_w)$, an assumption rather common in literature.

#### Van Genuchten Capillary Pressure Function

Derivation of the capillary pressure function can be obtained experimentally through measurement (see [21]) or through an analytical derivation of functional relationship between capillary pressure and saturation, see [49]. Usually this model contains several parameters that have to be fitted with

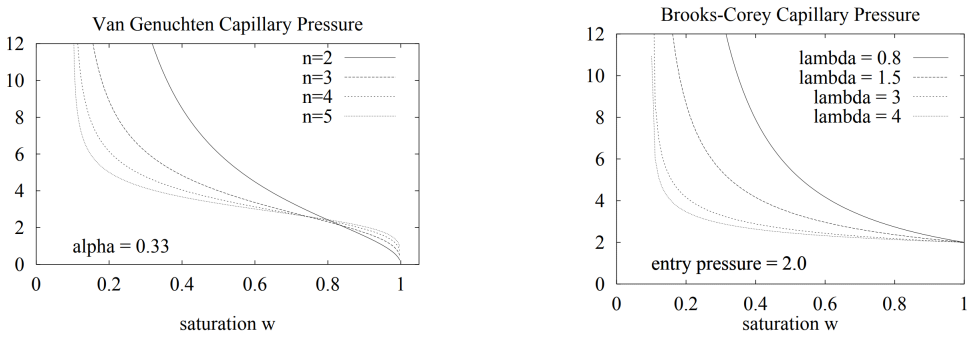


**Fig. 3.1.:** Van Genuchten and Brooks-Corey capillary pressure functions for different parameters (Figure from [6]).

experimental data. The Van Genuchten model is written in term of the *effective saturation* as

$$P_c(S_w) = \frac{1}{\alpha}\left(\bar{S_w}^{-\frac{1}{m}} - 1\right)^{1/n} \tag{3.4}$$

where the effective saturation is defined as

$$\bar{S_w} = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{nwr}},$$

$S_{wr}$ and $S_{nwr}$ are the wetting phase and non-wetting phase residual saturation, respectively. The parameter $m$ in (3.4), is often chosen as $m = 1 - \frac{1}{n}$ where $n$ is in the range $[2, 5]$ and $\alpha$ is related to the so called entry pressure, i.e. the minimal pressure required to force the non-wetting flow into a wetting rock during primary drainage.

### Brooks-Corey Capillary Pressure Function

Another model for a two-phase systems is given by the formula proposed by Brooks and Corey [12]

$$P_c(S_w) = p_d \bar{S_w}^{-\frac{1}{\lambda}} \tag{3.5}$$

where $p_d$ is the entry pressure of the porous medium and $\lambda$ is related to the pore size distribution, typical values of $\lambda$ are in the range $[0.2, 3]$. Fig. 3.1 shows the Van Genuchten function (left) and the Brooks-Corey function (right) for different values of $n$ and $\lambda$, respectively and fixed entry pressure.

### J-Leverett Capillary Pressure Function

Another commonly used mothel that describes the capillary pressure in a two-phase system in the J-Leverett function [65], given by

$$P_c(S_w) = \sigma \cos(\theta)\sqrt{\frac{\phi}{K}} J(S_w),$$

where $\sigma$ is the surface tension, $\theta$ is the contact angle, $\phi$ is the porosity, $K$ is the permeability and $J$ is the J-Leverett function:

$$J(S_w) = \gamma \left( \frac{S_w - S_{wi}}{1 - S_{wi}} \right)^{-0.5},$$

where $\gamma$ is normally chosen in the range $[0.05, 0.2]$ and $S_{wi} = 0.2$, see [67] for more details.

### 3.2.2 Relative permeability curves

In this subsection we review several laws that define the relative permeability $k_{r\alpha}$. In two-phase flow this leads to the functions proposed by Van Genuchten and Brooks-Corey.

#### Van Genuchten Relative Permeability

The Van Genuchten relative permeability functions for a two-phase system with wetting (w) and non-wetting phase (nw) are written in terms of residual saturation as

$$k_{rw} = \bar{S}_w^{\epsilon} \left( 1 - \left( 1 - \bar{S}_w^{\frac{n}{n-1}} \right)^{\frac{n-1}{n}} \right)^2,$$
$$k_{rnw} = \bar{S}_{nw}^{\gamma} \left( 1 - \left( 1 - \bar{S_{nw}} \right)^{\frac{n}{n-1}} \right)^{\frac{2(n-1)}{n}},$$

where $\epsilon$ and $\gamma$ are typically chosen as $\epsilon = \frac{1}{2}$ and $\gamma = \frac{1}{3}$, see [49]. Parameter $n$ is the same of eq. (3.4).

#### Brooks-Corey Relative Permeability

The Brooks-Corey model is given by the formulas:

$$K_{rw}(S_w) = \bar{S_w}^{-\frac{2+3\lambda}{\lambda}},$$
$$K_{rnw}(S_{nw}) = \bar{S}_{nw}^2 \left( 1 - \left( 1 - \bar{S}_{nw} \right)^{\frac{2+\lambda}{\lambda}} \right).$$

Also here, the parameter $\lambda$ is the same as in the capillary pressure function (3.5). Fig. 3.2 shows an example for the Van Genuchten relative permeability (left) and the Brooks-Corey relative permeability (right) for different values of $n$ and $\lambda$, respectively.

### 3.2.3 Immiscible two-phase flow: pressure-saturation formulation

In this work, we consider a two-phase flow system, the *wetting* and *non-wetting* phase. Under the incompressible fluid and solid assumption,

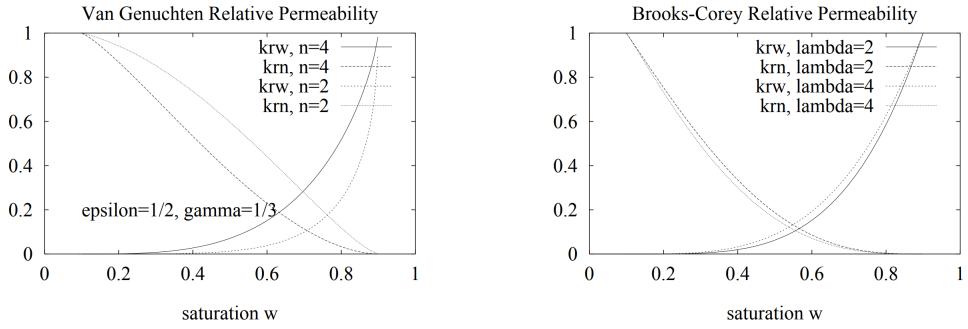


**Fig. 3.2.:** Van Genuchten and Brooks-Corey relative permeability functions for different parameters, residual saturations$S_{wr} = S_{nwr} = 1$ (Figure from [6]).

considering the non-wetting pressure and the wetting saturation as primary variable [16], the following pressure equation is obtained:

$$\nabla \cdot \mathbf{u}_t = q_t \qquad \text{in } \Omega, \tag{3.6}$$

$$\mathbf{u}_t = -\mathbf{K}\left(\lambda_t \nabla p - \lambda_w \nabla P_c - (\lambda_w \rho_w + \lambda_{nw}\rho_{nw}) g \nabla z\right). \tag{3.7}$$

where $p = p_{nw}, S = S_w$, $\lambda_t = \lambda_w + \lambda_{nw}$ is the total mobility, $q_t = \dfrac{q_w}{\rho_w} + \dfrac{q_{nw}}{\rho_{nw}}$ is the total source term and $\mathbf{u}_t = \mathbf{u}_w + \mathbf{u}_{nw}$ is the total velocity.

The saturation equation is transformed into

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot \left\{ \mathbf{K} f_w \lambda_o (\nabla P_c + (\rho_{nw} - \rho_w) g \nabla z) + f_w \mathbf{u}_t \right\} = q_t, \qquad \text{in } \Omega. \tag{3.8}$$

where $f_w = \lambda_w / \lambda_t$ is the fractional flow. Equations (3.6), (3.7) and (3.8) are coupled by the total velocity, the phase relative permeability $k_{r_\alpha}$ and the capillary pressure $P_c$.

## 3.3 Solution strategies

There are three main solution strategies to solve the equations described in the previous section: a fully implicit (or fully coupled) approach, the IMPES (IMplicit Pressure, Explicit Saturation) approach and the sequential implicit strategy.

### 3.3.1 Fully implicit method (FIM)

The FIM approach solves pressure and saturation equations simultaneously [77]. The equations are first discretized, both in space and time, and written in the residual form:

$$\mathsf{r}(x) = 0$$

where r is the residual and $x$ is the vector of the primary unknowns $(p, S_1, \ldots, S_{N_{p-1}})$. The residual is a nonlinear function of the unknowns so a

linearization technique – namely the Newton-Raphson method – is used to solve the equation:

$$\mathsf{r}^{\nu+1} \approx \mathsf{r}^{\nu} + \left.\frac{\partial \mathsf{r}}{\partial x}\right|^{\nu} \delta x^{\nu+1} = 0$$

$\nu$ is the iteration index and $\delta x$ is the increment. At each iteration, until convergence is achieved – i.e. residual smaller than a given tolerance – a linear system is solved

$$\mathsf{J}^{\nu} \delta x^{\nu+1} = -\mathsf{r}^{\nu}, \tag{3.9}$$

$\mathsf{J} = \dfrac{\partial \mathsf{r}}{\partial x}$ is the Jacobian matrix. At each iteration the linear system (3.9) has to be solved, it can be used a direct method or, if the system is too large, an iterative method [85].

### 3.3.2 Sequential approaches

Sequential approaches consist in solving the equations separately. First, the pressure equation (3.6) is solved. Then, the total velocity is calculated. And, at the end, the saturation equations are solved using eq. (3.8). In this way, all the transport dependent terms are given, since the pressure is know.

The simplest approach to solve these splitted equations is the IMPES method [4]. It employs an implicit time integration scheme for the pressure (it is a parabolic equation in the compressible case) and an explicit one to solve the saturation equations. In this way, the computational cost to solve each step is reduced. However, if we solve the saturation equation with an explicit scheme we may have a severe restriction of the time step because of the CFL condition. For this reason it is usually preferable to use an implicit time integration technique also for the saturation equations. The resulting procedure is called Sequential IMplicit (SIM) strategy. At each time step $\Delta t$ from current time $t^n$ to the next simulation time $t^{n+1}$, the solution at time $t^{n+1}$ is found by first solving the pressure equation, keeping all saturation dependencies frozen at their values at $t = t^n$, i.e.,

$$-\nabla \cdot \left(\mathbf{K}\lambda_t^n \nabla p^{n+1}\right) = q_t - \nabla \cdot (\mathbf{K}(\lambda_w^n \nabla P_c^n + (\lambda_w^n \rho_w + \lambda_{nw}^n \rho_{nw}) g \nabla z)) \,. \tag{3.10}$$

Then, the total velocity is computed as

$$\mathbf{u}_t^{n+1} = -\mathbf{K}\left(\lambda_t^n \nabla p^{n+1} - \lambda_w^n \nabla P_c^n - (\lambda_w^n \rho_w + \lambda_{nw}^n \rho_{nw}) g \nabla z\right),$$

and, finally, the new saturation values are found by solving,

$$\phi \frac{S^{n+1} - S^n}{\Delta t} + \nabla \cdot \left(\mathbf{K} f_w^{n+1} \lambda_o^{n+1} \left(\nabla P_c^{n+1} + (\rho_{nw} - \rho_w) g \nabla z\right) + f_w^{n+1} \mathbf{u}_t^{n+1}\right) = q_t, \tag{3.11}$$

which corresponds to an implicit Euler step. Sometimes, in cases with strong compositional effects (out of the topic of this thesis) this approach can lead to some instability of the solution. For this reason the sequential fully- implicit

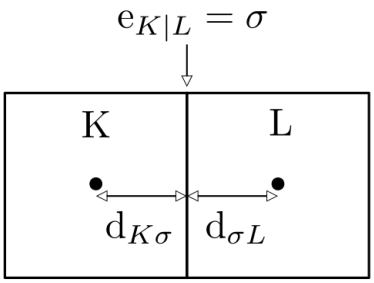


**Fig. 3.3.:** Example of control volumes $(K, L)$.

(SFI) method has been introduced [71, 72]. In this strategy sub-iterations between pressure and saturation equations are performed until convergence.

## 3.4 Implicit Finite Volume Discretization

The fine-scale (in space) discrete system is obtained by using a finite-volume-based two-point-flux-approximation scheme. Let $\mathcal{T}_h$ be the fine scale mesh with rectangular or hexahedral control volumes $K \in \mathcal{T}_h$ . The set of faces $e$ of generic element $K$ is denoted by $\mathcal{E}_K$. The flow equation (3.10) is discretized as

$$-\sum_{e_{KL}\in\mathcal{E}_K} P_{e_{KL}} = |K| q_{t_K} - \sum_{e_{KL}\in\mathcal{E}_K} B_{e_{KL}} \qquad \forall K \in \mathcal{T}_h, \tag{3.12}$$

where $|K|$ stands for the measure (volume for three dimensional elements, area in two dimensional elements) of element $K$, $e_{KL}$ is the face of element $K$ shared with element $L$, and the numerical flux for the pressure $P_{e_{KL}}$ is given by

$$P_{e_{KL}} = \tau_{K|L} \lambda^{UP}_{t_{KL}} (p_L^{n+1} - p_K^{n+1}) \qquad \forall e_{KL} \in \mathcal{E}_K,$$

where $p_K$ is the constant pressure approximation in the current cell $K$ and $p_L$ is the pressure value in the neighboring cell. The interface trasmissibility value $\tau_{K|L}$ is the harmonic average of the neighboring cell parameters [6], i.e.,

$$\tau_{K|L} = \frac{|e_{K|L}|}{\frac{d_{K\sigma}}{\mathbf{K}_K} + \frac{d_{\sigma L}}{\mathbf{K}_L}},$$

where $|e_{K|L}|$ is the area of the interface (length in two-dimensional domains), $d_{K\sigma}$ is the distance between the center of the current cell $K$ and the interface $\sigma = e_{K|L}$ and similarly, $d_{\sigma L}$ is the distance between the interface and the center of the cell $L$, as shown in Fig. 3.3.

The discrete phase mobility at the interface is given by:

$$\lambda^{UP}_{\alpha_{KL}} = \begin{cases} \lambda_\alpha(S^n_{\alpha_U}) & \text{if } \tau_{K|L}(p_K - p_L) + \tau_{K|L}\rho_\alpha g(z_L - z_K) > 0 \\ \lambda_\alpha(S^n_{\alpha_U}) & \text{otherwise,} \end{cases} \tag{3.13}$$

where, $S_U$ and $S_D$ denote the upstream and downstream saturation values, respectively. If, for example, the velocity propagates towards the right, the left

side of the cell $K$ is called upstream side and the right side is the downstream side, as described in [98].

The numerical fluxes at the right hand side are computed as follows:

$$\begin{aligned} B_{e_{KL}} =& \tau_{K|L}\lambda^{UP}_{w_{KL}}(P^n_{c_L} - P^n_{c_K}) + \\ &+\tau_{K|L}\left(\lambda^{UP}_{w_{KL}}\rho_w + \lambda^{UP}_{nw_{KL}}\rho_{nw}\right)(z_L - z_K) \qquad \forall e_{KL} \in \mathcal{E}_K, \end{aligned}$$

where the discrete phase mobilities are chosen as in (3.13). The residual of the discrete saturation equation (3.11) reads

$$\mathsf{r}^{n+1}_K = \phi(S^{n+1}_K - S^n_K) - \frac{1}{|K|}\sum_{e_{KL}\in\mathcal{E}_K} F^{n+1}_{e_{KL}} - q^{n+1}_{t_K} = 0. \tag{3.14}$$

Here, $F^{n+1}_{e_{KL}}$ is the numerical flux and it can be decomposed in three parts to have a separate treatment of the mobilities [66]:

$$F^{n+1}_{e_{KL}} = \left(V^{n+1}_{e_{KL}} + G^{n+1}_{e_{KL}} + C^{n+1}_{e_{KL}}\right)$$

$V^{n+1}_{e_{KL}}$ is the viscous numerical flux, $G^{n+1}_{e_{KL}}$ is the buoyancy numerical flux and $C^{n+1}_{e_{KL}}$ the capillary numerical flux. In the viscous numerical flux the mobilities are upwinded using the sign of the total velocity:

$$F^{n+1}_{e_{KL}} = \begin{cases} \Delta t f_w(S^{n+1}_U)u_{T_{KL}} & \text{if } u_{T_{KL}} > 0 \\ \Delta t f_w(S^{n+1}_D)u_{T_{KL}} & \text{otherwise.} \end{cases}$$

In the gravity numerical flux, the saturations for the mobilities are chosen based on the density differences:

$$G^{n+1}_{e_{KL}} = \begin{cases} \Delta t \tau_{K|L} f_w(S^{n+1}_U)\lambda_o(S^{n+1}_U)(\rho_{nw} - \rho_w)g(z_L - z_K) & \text{if } (\rho_{nw} - \rho_w)g(z_L - z_K) > 0 \\ \Delta t \tau_{K|L} f_w(S^{n+1}_D)\lambda_o(S^{n+1}_D)(\rho_{nw} - \rho_w)g(z_L - z_K) & \text{otherwise.} \end{cases}$$

Finally, the capillary numerical flux is approximated as:

$$C^{n+1}_{e_{KL}} = \begin{cases} \Delta t \tau_{K|L} f_w(S^{n+1}_U)\lambda_o(S^{n+1}_{w_U})(P^{n+1}_{c_L} - P^{n+1}_{c_K}) & \text{if } u_{T_{KL}} > 0 \\ \Delta t \tau_{K|L} f_w(S^{n+1}_D)\lambda_o(S^{n+1}_{w_D})(P^{n+1}_{c_L} - P^{n+1}_{c_K}) & \text{otherwise.} \end{cases}$$

The fractional flow, as the mobilities and the capillary pressure, are a nonlinear function of the saturation, and the Newton method is used to solve the nonlinear transport equation. Note that the flux function is non-convex, therefore, the Newton method may diverge for big time steps. To ensure convergence of the nonlinear method, a modified Newton approach has been used. The technique relies on the factor that for hyperbolic problems with S-shaped flux functions, typical of multi phase transport flows, the iteration of the Newton method is always converges if the initial guess is equal to the inflection point of the flux function $f(S)$ i.e. the point $S_C$ such that $f''|_{S_C} = \frac{d^2 f}{dS^2}|_{S_C} = 0$. However, in some practical cases it is difficult to determinate the location of the inflection point. Thus, modified Newton

works in the following way: at the end of each Newton iteration, if the sign of $f''$ at $S^{n+1,\nu}$ and $S^{n+1,\nu+1}$ (the superscripts $\nu$ and$\nu+1$ indicate the previous and the current iteration of the Newton iterations, respectively) are not the same; $S^{n+1,\nu+1}$ is replaced by $(S^{n+1,\nu+1}+S^{n+1,\nu})/2$ otherwise $S^{n+1,\nu+1}$ remains unchanged. More details may be found in [55].

### 3.4.1 Treatment of wells

In this work wells are modeled using Peaceman correction [78]. This is a standard way to treat a well as a source, or sink, term added to the gridblocks penetrated by the well [4].

Since well pressure is given, injection or production rate for each phase is expressed by

$$q_\alpha = WI\lambda_\alpha\left(p_\alpha - p^w\right), \tag{3.15}$$

where $WI$ is the productivity well index, $\lambda_\alpha$ is the already defined phase mobility, $p_\alpha$ is the phase pressure in the perforated gridblock and $p^w$ is the given wellbore pressure. In equation (3.15), the productivity well index describes the transmissibility between the wellbore and the perforated cell is modelled by

$$WI = \frac{2\pi\Delta z}{\ln\dfrac{r_b}{r_w}}$$

where $\Delta z$ is the perforated cell height, $r_b$ is the perforated cell radius and $r_w$ is the well radius.

# 4 A conservative multirate multiscale method for simulation of multiphase flow in porous media

In this chapter, we present a novel conservative multirate multiscale method for space-time conservative multiscale simulation of sequentially coupled flow and transport equations. First, the pressure is solved with the multiscale finite volume method. The iterative multiscale procedure is applied to guarantee the desired accuracy of the pressure solution. Once a good-approximate multiscale pressure solution is obtained, a conservative velocity field is constructed by solving local pressure equations subject to Neumann flux from the multiscale pressure solution. This conservative velocity field is then used to solve the saturation equation using a multirate method with a given accuracy tolerance. The multirate method employs, initially, a coarse-scale time step everywhere in the domain to estimate the updated saturation field. Then, based on the error estimate criterion, the location of high sensitivity regions (fast dynamics) is detected and solved with a smaller time step. The integration of the refined time-step zone (fast dynamics) and the rest of the domain (slow dynamics) is done via the flux-constrained formulation described in Chapter 2 that guarantees local mass balance. This combination of a space-time multiscale for flow and transport allows for reducing the computational cost without compromising the accuracy of the solution.

## 4.1 State of art

To tackle the simulation complexity with respect to the space, multiscale finite element [52, 32, 33] and finite-volume [56, 47] methods have been developed to construct a spatial coarse-scale systems for elliptic [1, 15, 58, 75] and parabolic [43, 18] flow equations with fine-scale heterogeneous coefficients. This is achieved by introducing locally-computed basis functions. After solving the coarse scale system, the approximate fine-scale solution can be found by interpolating the coarse-scale solution with the local basis functions. The multiscale method also allows converging to the fine-scale solution using the possibility of a conservative velocity construction at each iteration step [47]. It has been successfully integrated with both sequential [60] and fully implicit [25] coupling approaches. Among many other developments, recent advances include extensions to compositional [63, 44, 75, 76, 24] and geothermal flows [79, 96]. Multiscale simulation of fractured heterogeneous porous media have been also considered for both finite-volume and finite-element methods [45, 89, 2, 10, 95, 87]. An important feature of the

Parts of this chapter have been published in the Journal of Computational Physics [29]

multiscale procedure is that it can be formulated algebraically, which allows for convenient integration with existing commercial simulators for two-level [68, 101, 99] and dynamic multilevel [51] simulations.

In sequential simulation of two-phase flow, once the pressure solution is obtained, a saturation equation needs to be solved. Due to the sharp saturation fronts, small time steps are often required to achieve the desired accuracy. This challenge can be solved by the multirate techniques prevously described, which achieve the required accuracy with employing the refined time step only locally. They are different, yet complementary, to Adaptive Implicit (AIM) [90, 19, 69, 38, 39, 73] and potential ordering [61] methods, which aim at increasing stability by combining explicit-implicit integration strategies, as already explained in Chapter 1. We recall that, instead, in multirate methods the system is subdivided into two subsystems, one containing the active (fast) components that need a refined time step, and the other formed by the latent (slow) components, where the current time step is sufficient for the desired accuracy. Of particular interest is their application to heterogeneous porous media and their integration within nonlinear multiphase flow simulators.

## 4.2 Multirate multiscale approach

For efficient and accurate solution of the flow and the transport equations, i.e., Eqs. (3.12) and (3.14), respectively, we propose a multirate multiscale method which integrates multiscale techniques with our conservative approach. Fig. 4.1 provides an algorithmic overview of the method for one global time step. As illustrated, the multiscale finite volume method is used to solve the pressure equation, and the multirate method for the transport equation. More details will be provided in the following sub-sections.

### 4.2.1 Multiscale method for flow

The multiscale method considers two sets of coarse grids, denoted as primal and dual coarse grids, built starting from a given fine-scale computational grid. The primal coarse grid defines the control volumes to solve the pressure equation at the coarse-scale. The dual grid, on the other hand, provides the local supports for the computation of multiscale basis functions. Fig. 4.2 illustrates the construction of the coarse grids. Other special local functions, for instance well-functions near fine-scale source terms, can also be introduced in the dual coarse grids [99].

Let $M$ and $N$ be the number of coarse cells and of dual coarse cells, respectively. The multiscale method provides an approximation $p'$ of the fine-scale solution $p_f$ using a linear combination of solutions at the coarse grid nodes, namely

$$p_f(\mathbf{x}) \approx p'(\mathbf{x}) = \sum_{k=1}^{M} \Phi_k(\mathbf{x}) \bar{p}_k.$$

Here, $\Phi_k$ and $\bar{p}_k$ are respectively the basis function and coarse-scale solution associated to the coarse node $k$. The former is found by assembling specially

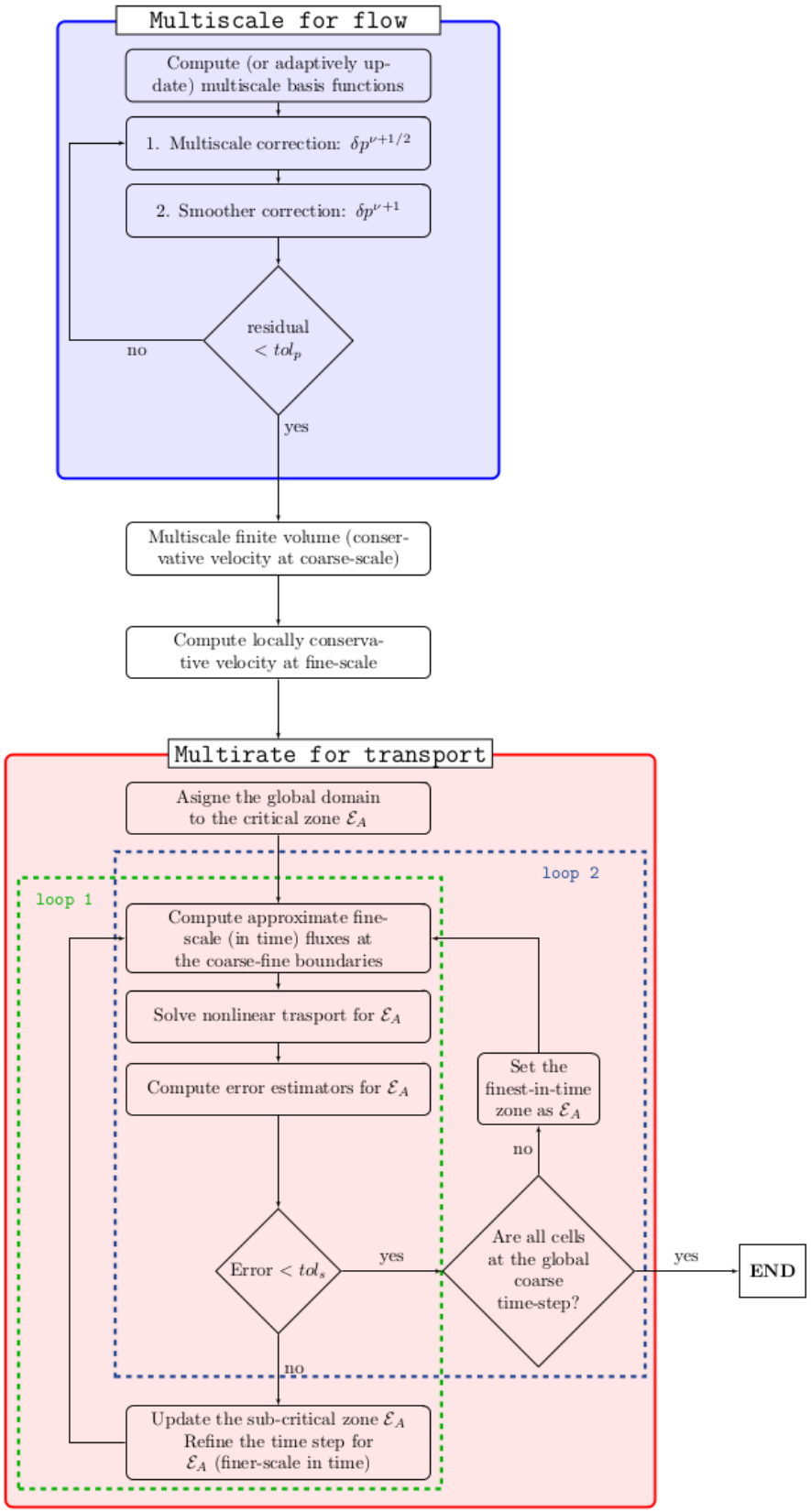


**Fig. 4.1.:** Description of the multirate multiscale algorithm for a time step.

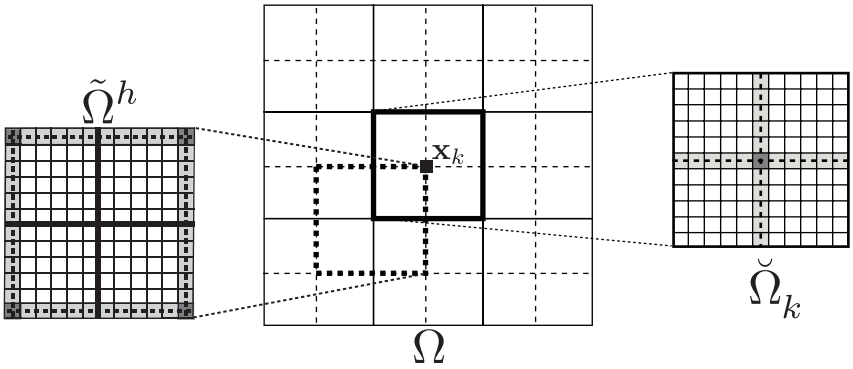


**Fig. 4.2.:** Illustration of the multiscale grids. Shown on the right and left are a coarse cell $\breve{\Omega}_k$ and a dual-coarse cell $\tilde{\Omega}^h$, respectively.

constructed local basis functions associated to that node in the dual coarse cells. More precisely, we set $\Phi_k(\mathbf{x}) = \sum_{h=1}^{N} \Phi_k^h(\mathbf{x})$, where each $\Phi_k^h(\mathbf{x})$ is built by solving

$$\begin{cases} -\nabla \cdot \left(\lambda_t \mathbf{K} \cdot \nabla \Phi_k^h\right) = 0 & \text{in } \tilde{\Omega}^h \\ -\nabla_{||} \cdot \left(\lambda_t \mathbf{K} \cdot \nabla \Phi_k^h\right)_{||} = 0 & \text{on } \partial\tilde{\Omega}^h \\ \Phi_k^h(\mathbf{x}_i) = \delta_{ik} & \forall \mathbf{x}_i \in \{1, ...M\} \end{cases}$$

on each dual-coarse cells. Here, subscript $||$ denotes a reduced problem along the tangential direction of the dual-coarse cell boundary. Note that, by construction, $\Phi_k^h$ is identically zero on coarse mesh cells not containing node $k$.

These basis functions can in fact be computed with some algebraic operations on the matrix of the fine-scale pressure system, as explained in [99], and collected as columns of the prolongation matrix P used to map the coarse-scale solution to the fine grid, i.e.

$$\mathsf{P} = \left[\Phi^1 \quad \Phi^2 \quad \dots \quad \Phi^N\right].$$

The restriction operator R, needed to map from fine to coarse-scale, is defined as

$$\mathsf{R}(i,j) = \begin{cases} 1 & \text{if fine cell } j \text{ is in coarse cell } i, \\ 0 & \text{otherwise.} \end{cases}$$

More details can be found in [99].

From the algebraic system of equations for pressure at fine-scale $Ap_f = b$, we build the operator $M_{MSFV}^{-1} = \mathsf{P}(\mathsf{R}A\mathsf{P})^{-1}\mathsf{R}$, and the iterative multiscale procedure: starting from a given $p^0$, for $\nu = 0, 1, \dots$ execute the following steps:

1. Multiscale stage: $\delta p^{\nu+\frac{1}{2}} = p^{\nu+\frac{1}{2}} - p^{\nu} = M_{MSFV}^{-1}\, r^{\nu}$;
2. Smoothing stage: $\delta p^{\nu+1} = p^{\nu+1} - p^{\nu+\frac{1}{2}} = M_{ILU(0)}^{-1}\, r^{\nu+\frac{1}{2}}$;

where the residual vector $r$ is updated using the latest available pressure, e.g. $r^{\nu+k} = b - Ap^{\nu+k}$. Steps 1 and 2 are repeated until the norm of the residual is below the desired threshold. Note that an ILU(0) factorization of the fine-scale matrix $A$ is adopted in the smoothing stage, as it has been found to be an effective choice [99]. In addition, before entering this iterative procedure, for the fine-scale source terms the initial guess $p^{\nu}$ is improved as

$$p^{\nu'} - p^{\nu} = M_W^{-1} r^{\nu},$$

where $M_W^{-1}$ is an adaptive local block-solver which acts only on dual-coarse cells with a nonzero fine-scale source terms. This can be seen as a form of well function [55]. Once the pressure is obtained, an additional MSFV stage is employed to obtain a coarse-scale conservative velocity field. Local pressure equations on the primal coarse cells are solved, subject to the velocity computed from the multiscale approximate pressure field. This stage produces a fine-scale locally conservative velocity which is then used to update the saturation equations [47].

### 4.2.2 Conservative multirate method for transport equation

We now present the details of the conservative multirate method for the efficient solution of the saturation equation (3.8). We recall that we are now facing a nonlinear equation whose solution may exhibit highly localized variations and fronts. The multirate method adopts different time-step sizes in different parts of the spatial domain to increase computational efficiency while preserving accuracy. The proposed method is based on a flux partitioning strategy to maintain local mass conservation.

The multirate method procedure can be summarized as follows: given the solution at time $t^n$, first an approximate solution at time $t^{n+1}$ is computed for all components using the coarse-scale time-step size $\Delta t$. This means that all cells will be initially assigned to the set of the critical zone $\mathcal{E}_A$, and, following the notation in Section 3.4, thus we compute

$$S_{\alpha K}^{n+1} = S_{\alpha K}^{n} - \frac{1}{\phi |K|} \sum_{e_{KL} \in \mathcal{E}_K} F_{e_{KL},\alpha}^{n+1} - q_{t_K}^{n+1}$$

The value of the numerical fluxes at all cell interfaces $e_{KL}$ is then checked, using an appropriate error estimator $\eta_{KL}$ that will be detailed later on. If a flux is rejected on the basis of the error estimator, i.e. if the error estimate is greater than a given tolerance, cells $K$ and $L$ are added to the new set $\mathcal{E}_A$. Accordingly, for each cell $K$ we can define the set of active fluxes (where the flux does not satisfy the error criterion) and the set of fluxes at the interface between fine and coarse regions, as

$$\mathcal{E}_{K_A} = \{\text{set of faces of the element } K : \eta_{KL} > \text{tol } \forall e_{KL} \in \mathcal{E}_K\},$$
$$\mathcal{E}_{K_L} = \{\text{set of faces of the element } K : \eta_{KL} \leq \text{tol } \forall e_{KL} \in \mathcal{E}_K\}.$$

Solution in the cells in $\mathcal{E}_A$ will be recomputed (locally) with a smaller time step, using an approximate discrete flux at the boundary, obtained by scaling the coarse-scale fluxes by the ratio of the new and the coarse time step. This approximation guarantees mass conservation at the global coarse time step. While, the saturation in cells that have a side at the interface between fine and coarse region is computed as

$$S_{\alpha_K}^{n+\frac{1}{2}} = S_{\alpha_K}^{n} - \frac{1}{\phi|K|}\sum_{e_{KL}\in\mathcal{E}_{K_L}}\frac{1}{2}F_{e_{KL},\alpha}^{n+1} - \frac{1}{\phi|K|}\sum_{e_{KL}\in\mathcal{E}_{K_A}}F_{e_{KL},\alpha}^{n+\frac{1}{2}} - q_{t_K}^{n+\frac{1}{2}}.$$

Here, $F_{e_{KL}}^{n+1}$ has been kept frozen at the value computed at the larger time step. If the new time step is such that all fluxes are accepted, the solution at time $t^{n+1}$ reads

$$S_{\alpha_K}^{n+1} = S_{\alpha_K}^{n+\frac{1}{2}} - \frac{1}{\phi|K|}\sum_{e_{KL}\in\mathcal{E}_{K_L}}\frac{1}{2}F_{e_{KL},\alpha}^{n+1} - \frac{1}{\phi|K|}\sum_{e_{KL}\in\mathcal{E}_{K_A}}F_{e_{KL},\alpha}^{n+1} - q_{t_K}^{n+1}.$$

The example above is a simple case where just one refinement took place. However, the method has been extended to deal with an arbitrary level of refinements until obtaining a desired flux quality. We have indeed adopted a self-adjusting strategy where the sub-critical zones are updated continuously until all fluxes satisfy the error threshold criterion. The method, therefore, entails two main loops. Loop 1 detects and integrates the sub-critical regions until the flux quality check is satisfied. The time-step in each new sub-critical zone is divided by 2. Loop 2 advances the sub-critical zones in time until the global time-step synchronization takes place. Fig. 4.3 illustrates a schematic example of how the two loops perform. The thick lines represent the sub-critical zones $\mathcal{E}_A$ and the highlighted sub-critical zones indicate cells where the transport equation has been solved either for refinement (Loop 1) or time advancing (Loop 2).

To check the flux quality we used an a-posteriori error estimator originally derived in [97]. It is based on the difference of fluxes at the current and previous local time-steps:

$$\eta_{KL}^{(i)} = |F_{e_{KL},\alpha}^{(i)} - F_{e_{KL},\alpha}^{(i-1)}|,$$

where superscript $(i)$ denotes here the $i$-th refinement inside the current global time step, while $(i-1)$ the previously accepted time step.

## 4.3 Numerical results

To test the performance of the multirate multiscale method, we consider the numerical test cases presented in [46]. In particular, the top and the bottom layers of the SPE10 test case are used to define the permeability field distribution [17].

We quantify the accuracy of the multirate multiscale solution by comparing it with the one obtained with fine-scale discretization both in space

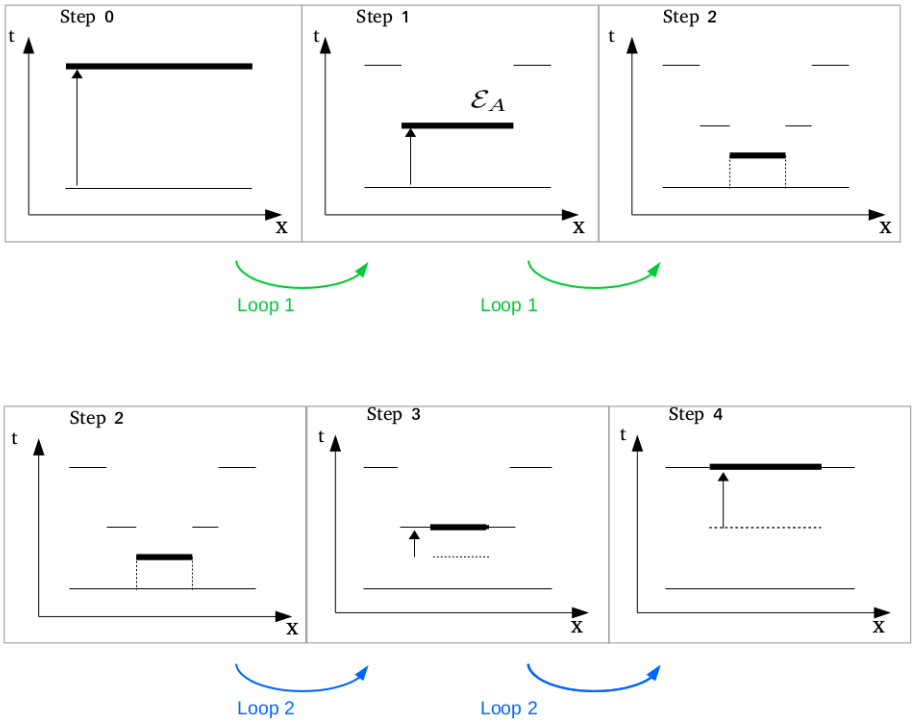


**Fig. 4.3.:** Schematic illustration of the integration in time with the multirate strategy for a global time step. In row (a) the Loop 1 has been applied to locally refine in time until the flux quality is satisfied. In row (b) Loop 2 is employed to advance the sub-critical zones in time until the global coarse time synchronization is reached. Note that Loop 1 and Loop 2 are fully integrated, meaning that for each step of Loop 2, Loop 1 will be called to maintain the flux quality.

and time (referred to as "reference solution" and denoted with the sub-index "ref"). Pressure relative errors and saturation absolute errors at each point and time are defined as

$$E_P(\mathbf{x},t) = \frac{|p(\mathbf{x},t) - p_{ref}(\mathbf{x},t)|}{|p_{ref}(\mathbf{x},t)|},$$

and

$$E_S(\mathbf{x},t) = |S(\mathbf{x},t) - S_{ref}(\mathbf{x},t)|,$$

respectively, where $|\cdot|$ is the absolute value.

We will also make use of the relative errors in $l^2$-norm at time $t$, given by

$$E_{2P}(t) = \frac{||p(t) - p_{ref}(t)||_2}{||p_{ref}(t)||_2}$$

and

$$E_{2S}(t) = \frac{||S(t) - S_{ref}(t)||_2}{||S_{ref}(t)||_2},$$

respectively. Here $||a||_2 = \sqrt{\sum_i a_i^2}$.

In the following test cases, we consider a rectangular domain and a fine-scale grid of $220 \times 60$ cells, over which the multiscale method imposes $20 \times 12$ coarse grid cells and two fluids with different viscosity. The less viscous fluid (water) is injected at the top-left cell $(1,1)$ with a non-dimensional rate of $10$ and the more viscous fluid (oil) is produced at the bottom-right cell $(220,60)$, which is kept at zero pressure. No-flow conditions are applied at the boundary in all test cases. In all simulations the domain is initially filled with oil, i.e. $S_o^0 = 1$. Quadratic relative permeability functions are employed, more precisely we set $k_w(S_w) = S_w^2$ and $k_o(S_w) = (1 - S_w)^2$. The viscosity ratio $\mu_o/\mu_w$ is set equal to $10$. In this section we neglected capillary and gravity effects.

### 4.3.1 Test case 1: SPE 10 top layer

We consider the top-layer permeability field of the SPE10 test case, as shown in Fig. 4.4. The simulation consider the time interval $(0, 600]$. For $t \in (0, 20)$, a time step size equal to $1$ is employed in order to start the multirate computation at $t = 20$ with a well developed saturation profile. The global (coarse-scale) time-step size during the multirate procedure is $\Delta t = 7.25$. The flux quality tolerance and the Newton iterative convergence threshold are set to $10^{-4}$ and $10^{-8}$, respectively. Fig. 4.5 shows the reference solution at final time $t = 600$.

As shown in Fig. 4.6, with the multirate approach only cells near the saturation front are solved at the fine timescale resolution. These cells are indeed associated with a fast process and sufficiently small time steps are needed to guarantee the required accuracy.

Fig. 4.7 illustrates pressure and saturation errors of simulations computed with coarse time steps and multirate (MR). For both of them, we have used the fine-scale grid in space. It is clear from this figure that the multirate

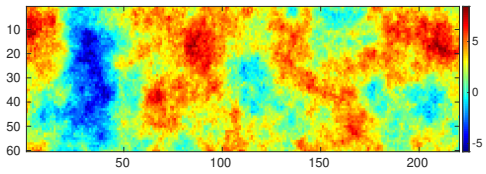


**Fig. 4.4.:** Test case 1 – Natural logarithm of the SPE10 top-layer permeability distribution.

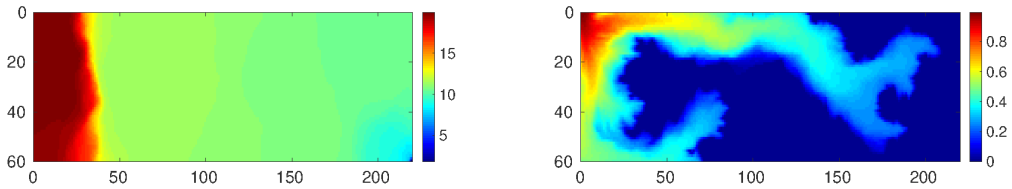


**Fig. 4.5.:** Test case 1 – Reference solution maps at final time $t = 600$ for the global pressure (left) and water saturation (right).

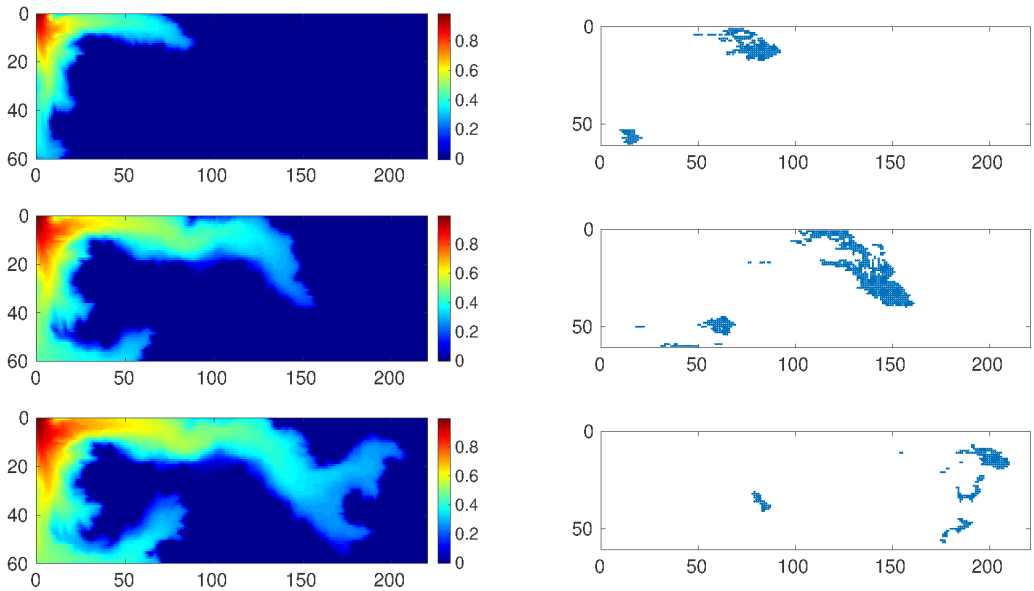


**Fig. 4.6.:** Test case 1 – Water saturation solution and active cells at times $t = 222.094$, $t = 435.062$ and $t = 599.094$.

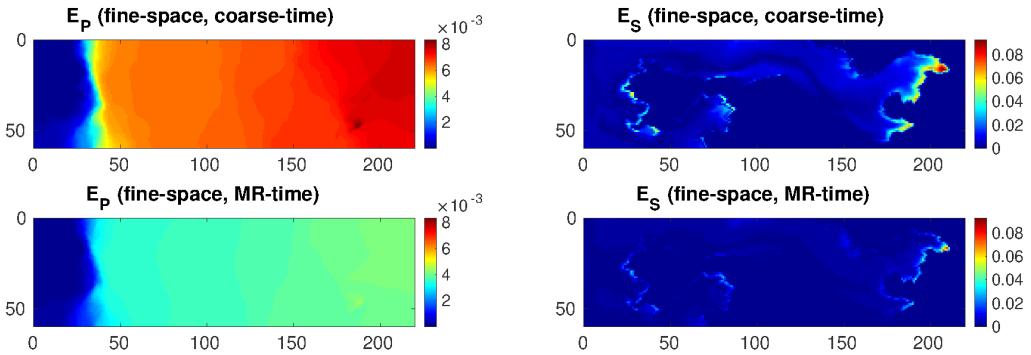


**Fig. 4.7.:** Test case 1 – Errors for the pressure (left column) and errors for the water saturation (right column) at final time $t = 600$ for the fine space grid, coarse time steps solution (top row) and for the fine space grid, multirate solution (bottom row).

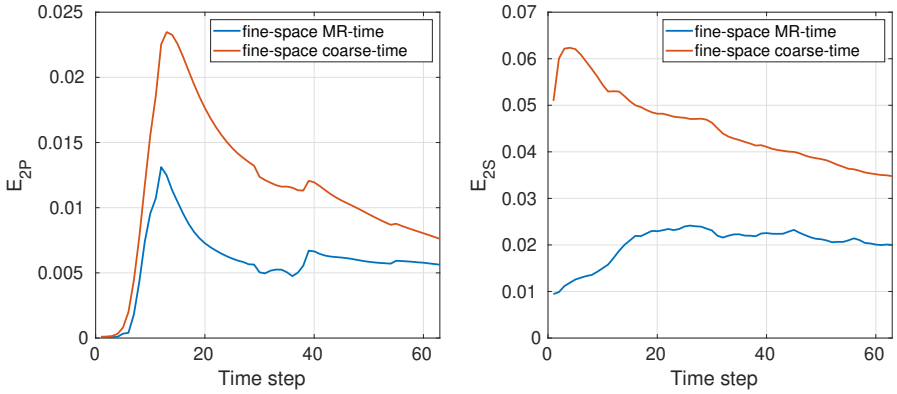


**Fig. 4.8.:** Test case 1 – Pressure (left) and saturation (right) relative errors in $l^2$-norm at each global step during simulation time for the multirate approach and using only coarse time steps, both computed using the fine space grid.

technique improves the solution taking only a small fraction of the cells at the fine-scale time step. Note that, since pressure and saturation equations are coupled, improving the saturation solution with the multirate method produces a more accurate pressure field.

Fig. 4.8 shows the $l^2$-norm of the error of the solutions obtained with the multirate approach and the coarse time steps approach at each global time steps (from time $t = 20$ to time $t = 600$). For both approaches, the solution from $t = 0$ to $t = 20$ is computed using small time steps to start the comparison from an already developed saturation field. The solution with overall coarse time steps start to accumulate errors immediately.

Fig. 4.9 shows that, compared with the single-rate fine timescale solver, the proposed multirate method applies more Newton iterations to converge to the nonlinear saturation solutions at each global (coarse-scale) time step. This fact motivates the definition of an indicator to estimate the overall computational cost as the cumulative sum of the number of Newton iterations times the number of active components. This indicator is indeed much lower for the multirate solution compared to that obtained in the single-

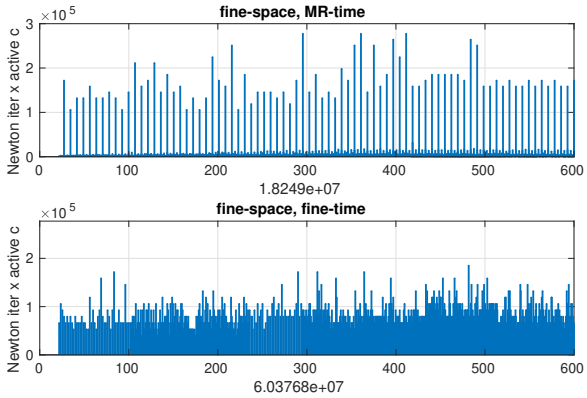


**Fig. 4.9.:** Test case 1 – Number of active cells multiplied by the number of Newton iterations at each time step (both global and local) for the multirate (MR) and fine-scale in time solvers. In the x-axis we show the corresponding simulation time. The value presented below each graphic is the total complexity of the respective solver.

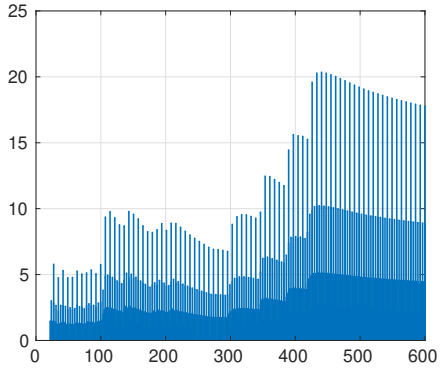

**Fig. 4.10.:** Test case 1 – CFL numbers (based on maximum analytical fractional flow derivative value) at each time step of the multirate method.

rate, fine time-step computation. This is because the steps performed with the multirate method with a large $\Delta t$ require more Newton iterations, as expected. However, this number drops quickly when the sub-critical zones are solved at smaller time step sizes because of local refinement. For this test case, the computational cost of the multirate solution is less than one third that of the fine-scale reference solver (the precise ratio is $0.29$).

Fig. 4.10 shows the CFL number associated with the adaptive time steps for multirate simulations, computed using the maximum value of the analytical flux derivative for the saturation. Large portions of the domain (far from the front) exhibit large CFL numbers, while a smaller fraction (near the front) advance with smaller CFL numbers. This illustrates the effectiveness of the proposed multirate method. In the simulations, the refinement of the time step is stopped once it leads to CFL $= 0.8$.

Now that the multirate method in time is fully investigated, we combine it with the multiscale method in space. Fig. 4.11 reports the errors of the

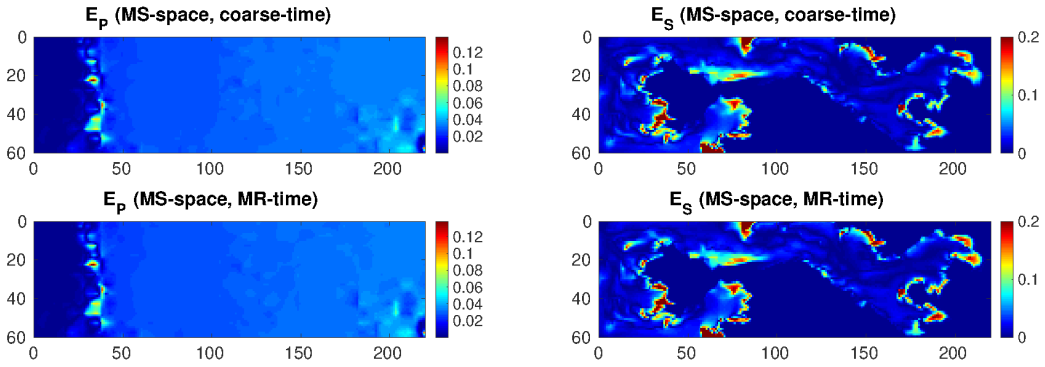


**Fig. 4.11.:** Test case 1 – Errors for the pressure (left column) and errors for the water saturation (right column) at final time for the multiscale with coarse time steps approach (top row) and for the multirate multiscale method (bottom row).

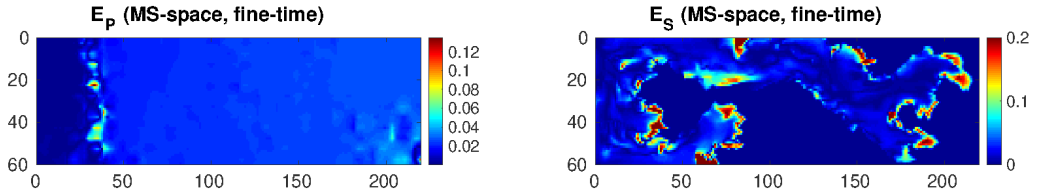


**Fig. 4.12.:** Test case 1 – Errors for the pressure (left column) and errors for the water saturation (right column) at final time for the multiscale with the fine time steps approach.

multiscale in space and coarse time-step (top) solution as well as those obtained with the multiscale multirate approach (bottom). There are no notable differences between the two solutions because the spatial errors introduced by the multiscale procedure dominate the overall errors. In fact, as shown in Fig. 4.12, the errors of the multiscale method with fine time steps are indistinguishable from those reported in Fig. 4.11.

Fig. 4.13 shows the simulation errors, similarly to Fig. 4.11, but with the iterative multiscale solver in space. Here, the two-stage multiscale solver is employed until the $l^2$-norm of the pressure residual is equal to $10^{-3}$ (top) and equal to $10^{-5}$ (bottom). For both the simulations the number of smoothing per iteration step was set to $n_s = 5$. Consistently with what reported in literature, decreasing the pressure residual tolerance brings more accurate solution for both pressure and saturation.

Fig. 4.14 presents the multirate multiscale errors compared with those of multiscale in space single-rate coarse-step in time. It is evident that the multirate method improves the solution by applying fine-scale time-steps only close to the saturation front. This is further elaborated in Fig. 4.15, which reports the $l^2$-norm of pressure saturation relative errors at each global time-step. Also in this case fine time-steps equal to $1$ are employed for both the approaches in the initial part of the simulation ($t \in (0, 20)$) to have a good initial saturation profile near the injection point.

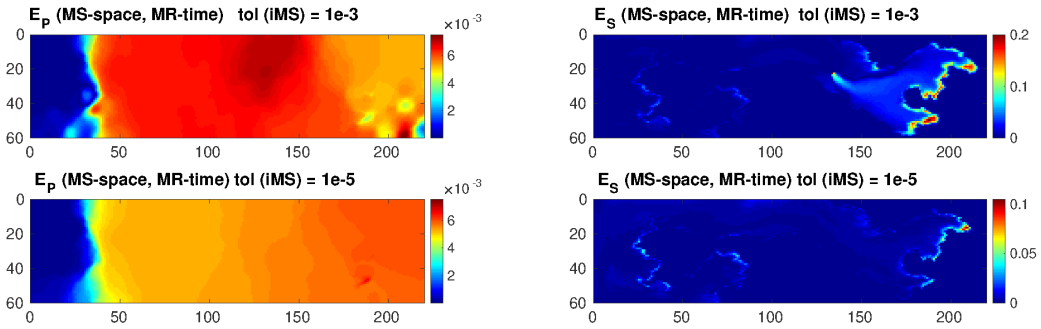


**Fig. 4.13.:** Test case 1 – Errors for the pressure (left column) and errors for the water saturation (right column) at final time for the multirate iterative multiscale approach with tolerance equal to $10^{-3}$(top row) and equal to $10^{-5}$ (bottom row).

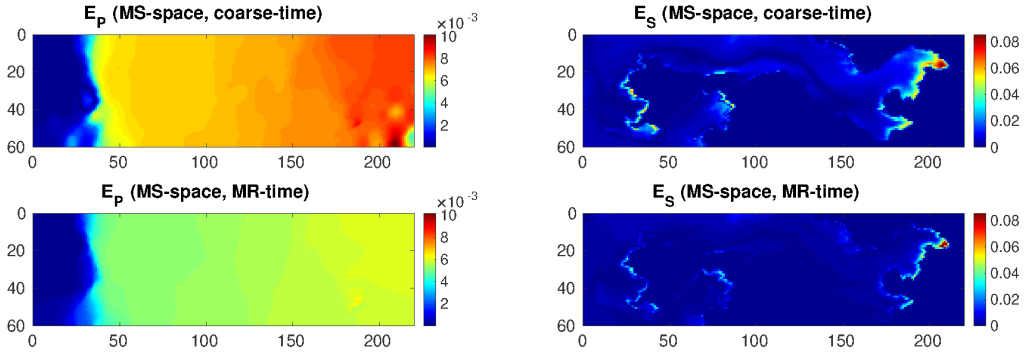


**Fig. 4.14.:** Test case 1 – Errors for the pressure (left column) and for the water saturation (right column) at final time for the coarse time steps (top row) and for the multirate method (bottom row), both considering the iterative multiscale approach with tolerance equal to $10^{-5}$ .

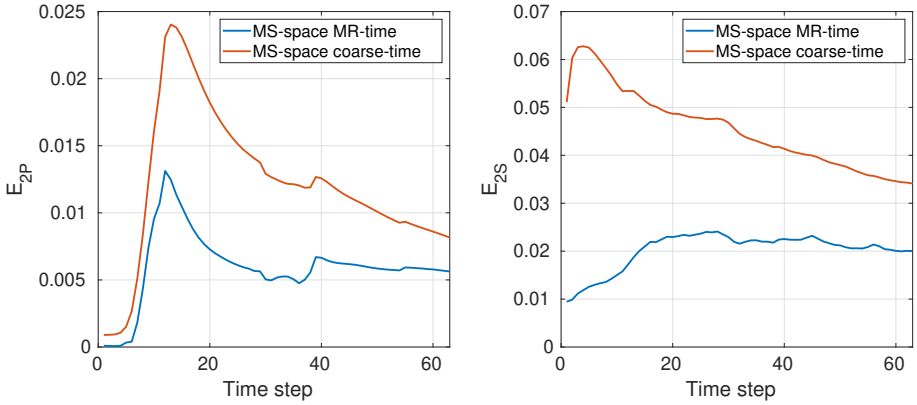


**Fig. 4.15.:** Test case 1 – Evolution of pressure (left) and saturation (right) relative errors in $l^2$-norm at each global time step for the multirate iterative multiscale approach and the iterative multiscale, coarse time steps approach. The iterative multiscale tolerance is $10^{-5}$.

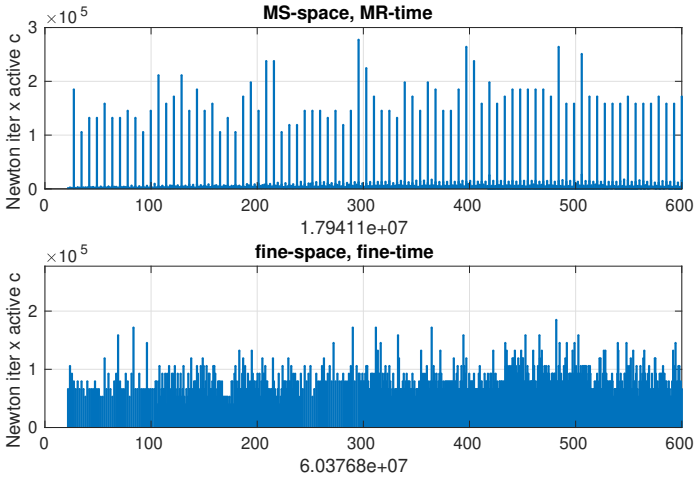


**Fig. 4.16.:** Test case 1 – Number of active cells multiplied by the number of Newton iterations at each time step for the multirate multiscale and reference solvers. The iterative multiscale tolerance is $10^{-5}$. The value presented below each graphic is the total complexity of the respective solver.

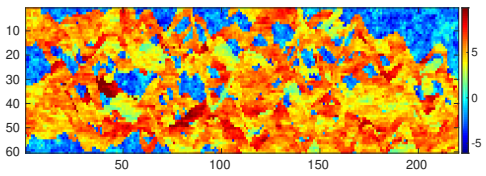

**Fig. 4.17.:** Test case 2 – Bottom-layer logarithmic permeability distribution.

Finally, the number of Newton iterations multiplied by the number of active cells (complexity) throughout the simulation time is shown in Fig. 4.16. We can notice that, even if the pressure profile is computed with the multiscale approach, the complexity of the transport equation is comparable with that of the previous test case, where we have adopted the fine-scale approach to compute the pressure.

### 4.3.2 Test case 2: SPE 10 bottom layer

In the second test case, we use the permeability field of the SPE 10 bottom layer, which has higher contrasts and more a channelized distribution, see Fig. 4.17.

As in the previously, the simulation starts at time $t = 0$ and ends at time $t = 600$. The global time step is equal to $5.5$ starting from time $t = 5$. In the range t=(0,5] a fine time step equal to $1$ has been employed for all approaches (multirate, fine time-steps and coarse time steps). The flux tolerance for the multirate approach is equal to $10^{-3}$ and the Newton convergence tolerance is $10^{-8}$. Fig. 4.18 shows the reference solution at final time.

Fig. 4.19 illustrates the active cells where the saturation transport equation is solved by the multirate method at a fine timescale, at three different instants. The method detects regions where the saturation front is moving fast and it is necessary to use smaller time steps.

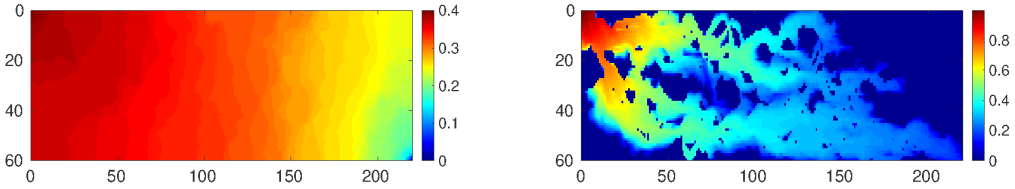


**Fig. 4.18.:** Test case 2 – Reference solution of pressure (left) and water saturation (right) at final time $t = 600$.

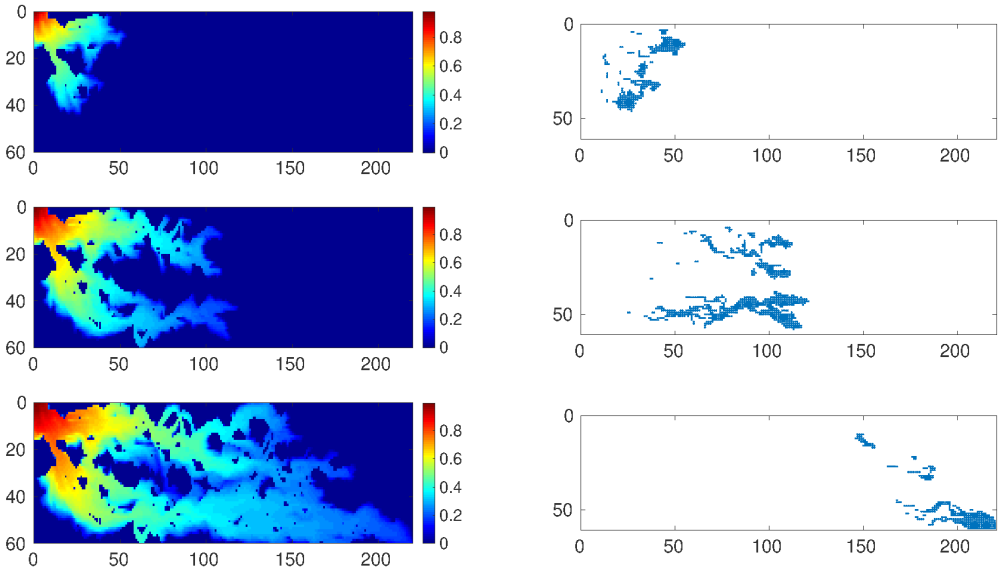


**Fig. 4.19.:** Test case 2 – Water saturation solution and active components at times $t = 103.176$, $t = 298.443$ and $t = 599.493$.

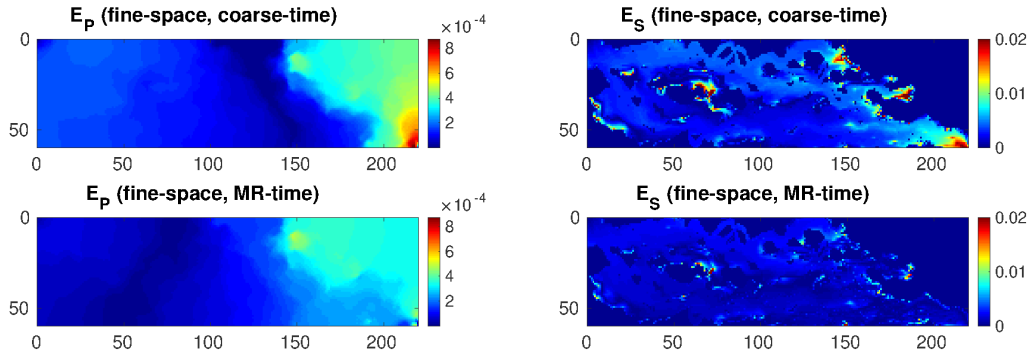


**Fig. 4.20.:** Test case 2 – Errors for the pressure (left column) and for the water saturation (right column) at final time for the coarse time steps solution (top row) and for that obtained with the multirate method (bottom row). Both simulation used the fine spatial grid.

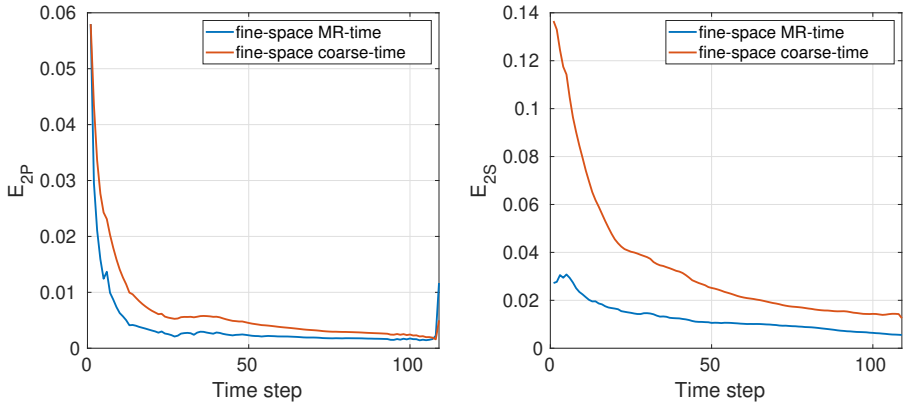


**Fig. 4.21.:** Test case 2 – Pressure (left) and saturation (right) relative errors in $l^2$-norm during the simulation time for the multirate approach and the coarse time steps approach, both considering the fine space grid.

As in the previous test case, the multirate approach is employed firstly on a fine spatial grid. Fig. 4.20 shows the errors for single-rate coarse time steps (top) and the multirate method (bottom). The multirate technique improves both the pressure error (left) and the saturation error (right). Note that for this challenging case, the coarse time-step solution does not capture the right saturation fronts accurately. The multirate method, instead, leads to a significantly improved solution by employing fine-scale time steps only in the vicinity of the front.

Fig. 4.21 shows the $l^2$-norm of pressure and saturation relative errors at each global time-step, starting from time $t = 5$, for both the multirate and overall coarse time-step approaches (using the fine spatial grid in both cases). As expected, the coarse time-step approach is less accurate than the multirate method.

Fig. 4.22 presents the computational complexity of the methods, i.e. the number of Newton iterations multiplied by the number of active cells, for simulations obtained with the multirate (top), fine time-step (center), and

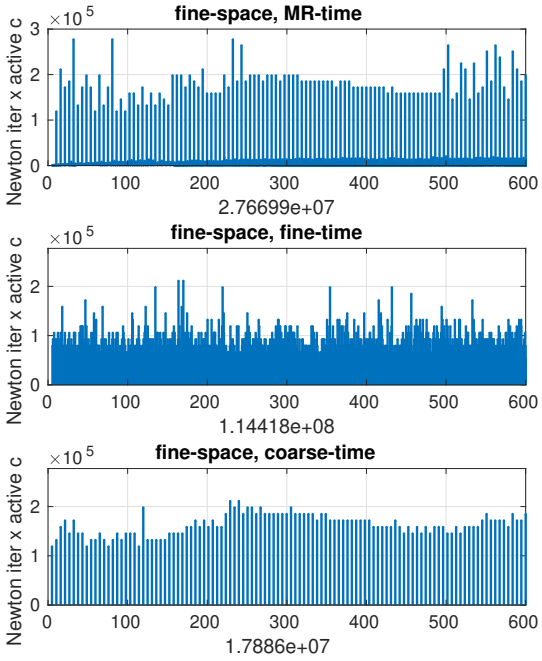


**Fig. 4.22.:** Test case 2 – Complexity of the problem: number of Newton iteration per active components required at each time step for the multirate (top), fine time steps (center) and coarse time steps (bottom) approaches. The value presented below each graphic is the total complexity of the respective solver.

coarse time-step (bottom) approaches. The ratio between the computational complexity of the multirate method and that of the reference (fine-scale in space and time) simulation approach is $0.24$.

The CFL number for the multirate method are given in Fig. 4.23. Also here, the CFL number is computed based on the maximum analytical fractional flow derivative value. Thus, clearly, coarse-scale (global) time steps have rather high CFL values.

Finally, the multirate multiscale method is investigated. The iterative multiscale solver tolerance is set to $10^{-6}$ with $n_s = 30$ ILU(0) smoothing steps per iterations. Note that we used the same size for the global time steps as the previous test case.

Fig. 4.24 illustrates the pressure and saturation errors for the solutions of the iterative multiscale in space and coarse-scale in time (top) and multirate in time (bottom) methods. The results are analogous to those obtained with the multirate approach and the coarse time steps approach where the fine-scale grid was applied. Fig. 4.25 shows the relative errors at the global time steps for the coarse-scale and multirate in time approaches, where both employ the iterative multiscale in space simulation approach. Both the multirate and the single-rate with coarse time steps are performed from time $t = 5$. Also, these results are analogous to the results obtained with the fine-scale grid in space. Finally, Fig. 4.26 shows the number of Newton

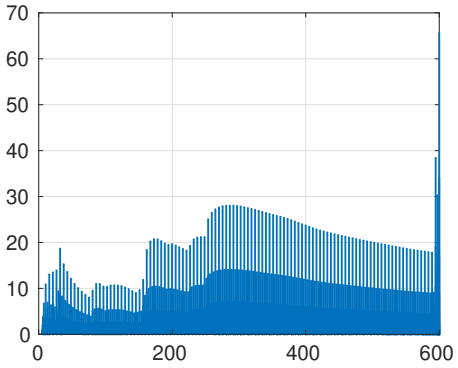


**Fig. 4.23.:** Test case 2 – CFL numbers at each time steps for the multirate in time (fine-scale in space) approach.

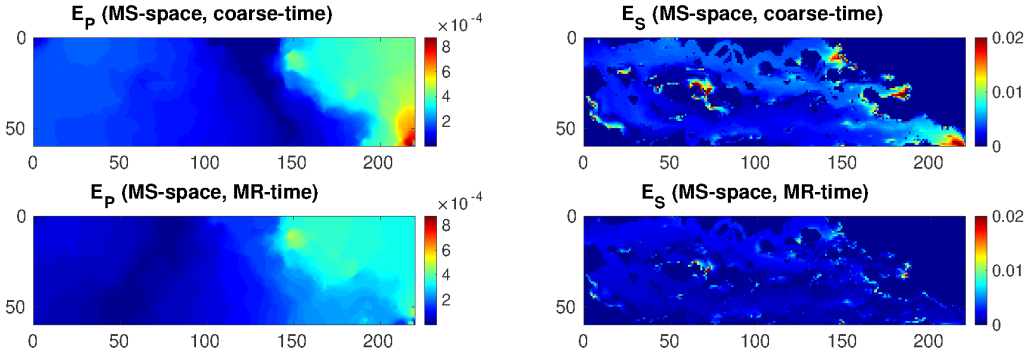


**Fig. 4.24.:** Test case 2 – Errors for the pressure (left column) and for the water saturation (right column) at the final time for the iterative multiscale coarse time steps approach (top row) and for the multirate iterative multiscale approach (bottom row).

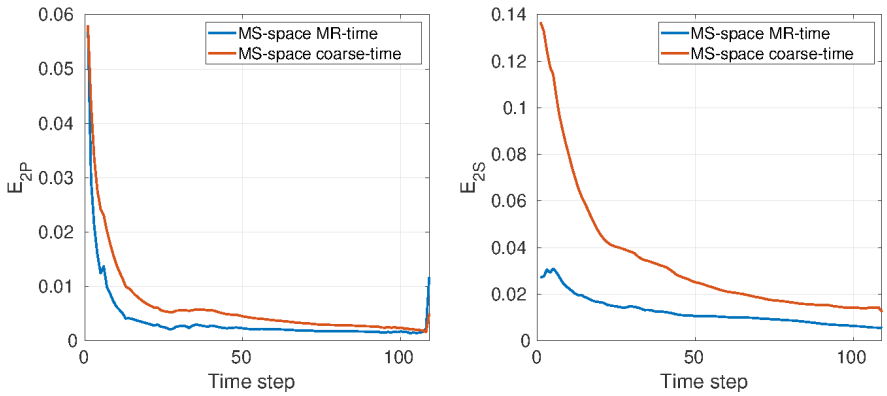


**Fig. 4.25.:** Test case 2 – Pressure (left) and saturation (right) relative errors in $l^2$-norm during the simulation time for the multirate iterative multiscale approach and the iterative multiscale coarse time steps approach.

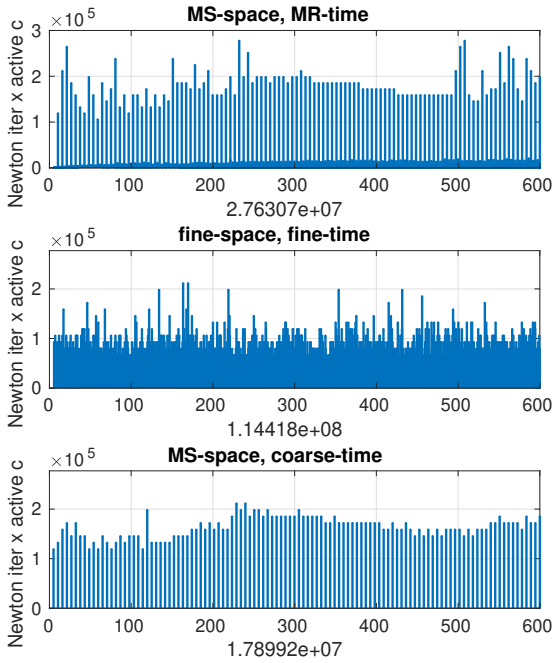


**Fig. 4.26.:** Test case 2 – Number of Newton iteration multiplied by the active cells at each time step for the multirate iterative multiscale approach (top), the fine-scale in time and space (center) and the iterative multiscale coarse-scale in time (bottom) approaches. The value presented below each graphic is the total complexity of the respective solver.

iterations multiplied by the active (critical sub-domain) cells for the three different approaches. As in the previous test case, the multirate method reduces the complexity of the solution obtained from the transport equation by a factor equal to $0.24$. The error introduced by the multiscale method to solve the pressure equation does not affect the performance of the multirate approach.

# 5 Algebraic dynamic multilevel method with local time-stepping (ADM-LTS) for sequentially coupled porous media flow

This chapter presents an algebraic dynamic multilevel method with local time-stepping (ADM-LTS) applied to the equation describing two-phase flow in heterogeneous porous media. The method employs an adaptive multilevel space-time grid determined on the basis of two error estimators, one in time and one in space. More precisely, at each time step, first a coarse time step on the coarsest space-grid resolution is computed. Then, based on the error estimators, the transport equation is solved by taking different time step sizes at different spatial resolutions within the computational domain. In this way, the method is able to use a fine grid resolution, both in space and in time, only at the moving saturation fronts. In order to ensure local mass conservation, two procedures are developed. First, finite-volume restriction operators and constant prolongation (interpolation) operators are developed to map the system across different space-grid resolutions. Second, the fluxes at the interfaces across two different time resolutions are approximated with the same averaging scheme in time used for the multirate approach and described in Chapter 2. Several numerical experiments have been performed to analyze the efficiency and accuracy of the proposed ADM-LTS method for both homogeneous and heterogeneous permeability fields. The results show that the method provides accurate solutions, and at the same time it reduces the number of fine grid-cells both in space and in time.

## 5.1 State of Art

As explained in Chapter 1, simulation of multiphase flow in natural porous media is challenging due to the variety of time and length scales involved in the process. Accurate numerical models require very high resolution both in space and time to capture all relevant physics. However, for large-scale simulations these kind of grids are unpractical. As already mention in Chapter 1, the computational cost is reduced by employing upscaling methods [31] which define effective rock and fluid properties to represent the relevant physics at a much coarser resolution. However, in presence of highly heterogeneous permeability fields and whenever a clear scale separation is not present, excessive upscaling may not give accurate results [26]. For this reason, advanced and scalable algorithms have to be developed to allow for efficient simulation on high resolution grids without having to define upscaled quantities.

Parts of this chapter have been published in MOX/Report No. 18/2019

Multiscale methods [52, 56, 47, 101, 75] and dynamic local grid refinement technique [7, 36] are among these advanced simulation strategies. The first ones have been already presented in details in Chapter 4. Instead, the latter takes advantage of the locality of transport processes by dynamically adapting the grid resolution so to allow for accurate transport simulations even in presence of sharp gradients (e.g., [94, 26]). Recently, the Algebraic Dynamic Multilevel (ADM) method [23, 26] has been developed to combine the consistent multilevel mapping of the pressure field throughout different grid resolutions with an adaptive grid refinement technique. In ADM simulation the governing equations are first discretized on a high resolution grid (referred to as fine-scale). Then, they are mapped and solved on a dynamically defined multilevel spatial grid system. The final solution can be provided both at the dynamic multilevel and fine-scale resolutions, through a sequence of prolongation and restriction operators.

Along with the advancements in the space-grid adaptivity, in order to reduce the overall simulation time, an implicit time integration scheme is usually employed to allow for much larger time step sizes (compared with the explicit alternative). The use of implicit integration, in the presence of strong non-linearities, requires Newton-Raphson iterations at each time step; however the Newton-Raphson method fails to converge for large time steps. As a consequence, several remedies have been proposed to enhance non-linear convergence [55, 100, 48] improving the numerical fluxes and the internal updates of the nonlinear loops; these allow for the use of very large time steps and considerably improve simulation time. However, the excessive numerical dispersion introduced by the use of large time steps can significantly impact the accuracy of the solution, by, for example, smearing the advancing saturation front. Thus, multirate or local time-stepping (LTS) approaches are of great interest for porous media flow simulation. These methods employ different time step sizes within the domain based on the local flow characteristics. In [54], an explicit adaptive conservative time integration techniques is presented, where the sizes of the local time steps are imposing by the CFL restriction.

## 5.2 ADM-LTS method

In this section, first, the original ADM method [23] is reviewed, then, the newly proposed ADM-LTS algorithm is presented in detail.

### 5.2.1 ADM method

The ADM method is employed to reduce the computational cost associated with the solution of the linear system arising from the linearized problem of Eq. (3.11).

Let us consider a domain discretized with a high resolution grid which is assumed to be fine enough to capture all relevant physics and to honour the heterogeneous distribution of the geological properties. Given this fine-scale discretization, a hierarchy of $n_l$ nested coarse grids is constructed. Each grid

is formed by $N_l = N_{lx} \times N_{ly} \times N_{lz}$ grid cells, where $l$ is the resolution index and $l = 0$ represents the fine grid resolution.

The set of all grid cells belonging to resolution level $l$ is called $\Pi^l$. At each time step ADM defines a multilevel grid by combining grid cells belonging to the hierarchy of grids previously defined. Given a multilevel ADM grid, let us define $\Omega^l$ as the set of grid cells belonging to all levels from 0 to $l$ which are present in the ADM grid. Additionally, it is convenient to define the set $\Gamma^l$ as $\Gamma^l = \Omega^l \cap \Pi^l$.

Eq. (3.14) is solved with a Newton-Raphson's method and at each Newton's iteration, a system of the form $\mathsf{J}^\nu \delta \mathsf{x}^{\nu+1} = -\mathsf{r}^\nu$ has to be solved. Here, $\delta\mathsf{x}$ is the vector of increment for saturation of phase $\alpha$, $\mathsf{J}$ is the Jacobian matrix and $\mathsf{r}$ is the residual.

Given an ADM grid formed by the set of grid cells $\Omega^l$, ADM assumes that the fine scale solution can be approximated by employing a sequence of prolongation operators, i.e.

$$\delta \mathsf{x}_{w_f} \approx \delta \mathsf{x}'_w = \hat{\mathsf{P}}^1_0 \dots \hat{\mathsf{P}}^l_{l-1} \delta \mathsf{x}^{ADM}_w.$$

Here, operator $\hat{\mathsf{P}}^i_{i-1}$ interpolates the solution at level $i$ to the finer resolution level $(i-1)$ and $\delta \mathsf{x}^{ADM}_w$ is the vector of increment for the ADM solution on the adaptive multilevel grid. $\delta \mathsf{x}_{w_f}$ is the increment vector at the fine scale.

The fine-scale Jacobian system is mapped to the ADM grid by

$$\hat{\mathsf{R}}^{l-1}_l \dots \hat{\mathsf{R}}^0_1 \mathsf{J} \hat{\mathsf{P}}^1_0 \dots \hat{\mathsf{P}}^l_{l-1} \delta \mathsf{x}^{ADM}_w = -\hat{\mathsf{R}}^{l-1}_l \dots \hat{\mathsf{R}}^0_1 \mathsf{r}_f, \tag{5.1}$$

where $\hat{\mathsf{R}}^{i-1}_i$ is the restriction operator which maps the solution from resolution at level $i$ to coarser level $(i-1)$. In order to ensure mass conservation at all levels, a finite volume restriction operator is considered [56]. Thus, the entry $(i, j)$ of a restriction operator reads

$$\hat{\mathsf{R}}^{l-1}_l(i,j) = \begin{cases} 1 & \text{if cell } i \in \Gamma^l \text{ and cell } j \in \Gamma^{l-1}, \\ \delta_{ij} & \text{otherwise.} \end{cases}$$

Additionally, constant interpolation is considered,

$$\hat{\mathsf{P}}^l_{l-1} = \left(\hat{\mathsf{R}}^{l-1}_l\right)^T.$$

### 5.2.2 ADM with local time-stepping (ADM-LTS)

At each time step $n$, after having solved the pressure equation and after having computed the total velocity field, the transport equation is solved employing the ADM-LTS algorithm.

First, Eq. (5.1) is solved over the whole domain on the coarsest grid resolution ($l_{max}$) formed by cells belonging to $\Pi^{l_{max}}$ and limiting refinement only around the wells, with time step $\Delta t$.

Based on the coarse solution obtained, the proper ADM grid resolution is chosen according to a front-tracking criterion. Two alternative front-tracking strategies are considered in this work:

- a criterion based on the saturation difference between neighbouring cells. A cell $i$ belonging to level $l$ is refined whenever the saturation difference, as defined in [23], between $i$ and one of its neighbours exceeds a user-defined tolerance $\epsilon_x$.

- a time-dependent criterion combined with the previous one to determine whether cells belonging to $\Pi^0$ should keep fine. Let us define $\psi_S = S^{n+1} - S^n$. A fine cell $i$ is kept at the fine resolution only if $\psi_{S_i} > \epsilon_t$, where $\epsilon_t$ is a user-defined tolerance. A similar time-based coarsening criterion has successfully been used in the literature for channelized heterogeneous problems where stationary gradients are present [22].

Once the ADM grid resolution has been defined, the solution is recomputed for all cells belonging to $\Omega^{l_{max}-1}$ with a time step $\Delta t_{l_{max}-1} = \Delta t/\eta$ by imposing local boundary conditions as described in detail in the following subsection. Here $\eta$ is the time refinement ratio. Then, the same operation is repeated for all resolution levels $l$ until $l = 0$ has been reached. Thus, each resolution level $l$ (formed by the set of grid blocks $\Omega^l$) is solved with a time step $\Delta t_l = \Delta t_{l+1}/\eta$. For the finest level ($l = 0$) ADM-LTS only recomputes the solution, with time step $\Delta t_0 = \Delta t/\eta^{l_{max}}$, for a subset, defined $\Omega^0_A$, of the cells belonging to $\Omega^0$. Only fine cells for which $\psi_S = S^{n+\Delta t_1} - S^n > \epsilon_t$ are part of the set $\Omega^0_A$.

The method advances in time for the active cells in $\Omega^0_A$ until they reach $t = t^n + \Delta t_1$. Once they are synchronized, cells in $\Omega^1$ advance in time. At this point, a new set of cells $\Omega^0_A$ is selected and these cells are advanced by $\Delta t_1$ performing $\eta\ \Delta t_0$ time steps. Once all cells in $\Omega^1$ have reached time $t = t^n + \Delta t_2$ another time step $\Delta t_2$ can be performed for all cells belonging to $\Omega^2$. This is a recursive procedure which is performed for all levels until all cells have reached time $t^{n+1} = t^n + \Delta t$.

Fig. 5.1 illustrates a schematic overview of the ADM-LTS method where $\eta$ and $l_{max}$ are both equal to 2. Fig. 5.2 shows an example of the ADM grid at each step and the refining area. At the global time step $\Delta t$, the solution is computed on the coarsest resolution $l_{max}$. At the intermediate time step the ADM grid resolution is defined and the solution is recomputed with the intermediate time step everywhere except in the coarsest region (middle figure). At the end, the method checks the errors and defines the set of active cells $\Omega^0_A$ (pink region on the right) where the solution is recomputed with the smallest time step.

### Local systems and local boundary conditions

For each resolution level $l$, the set of grid cells $\Omega^l$ is solved with the corresponding time step $\Delta t_l = \dfrac{\Delta t}{\eta^{(l_{max}-l)}}$. The number of active cells contained by $\Omega_l$ is denoted by $N^l_A$.

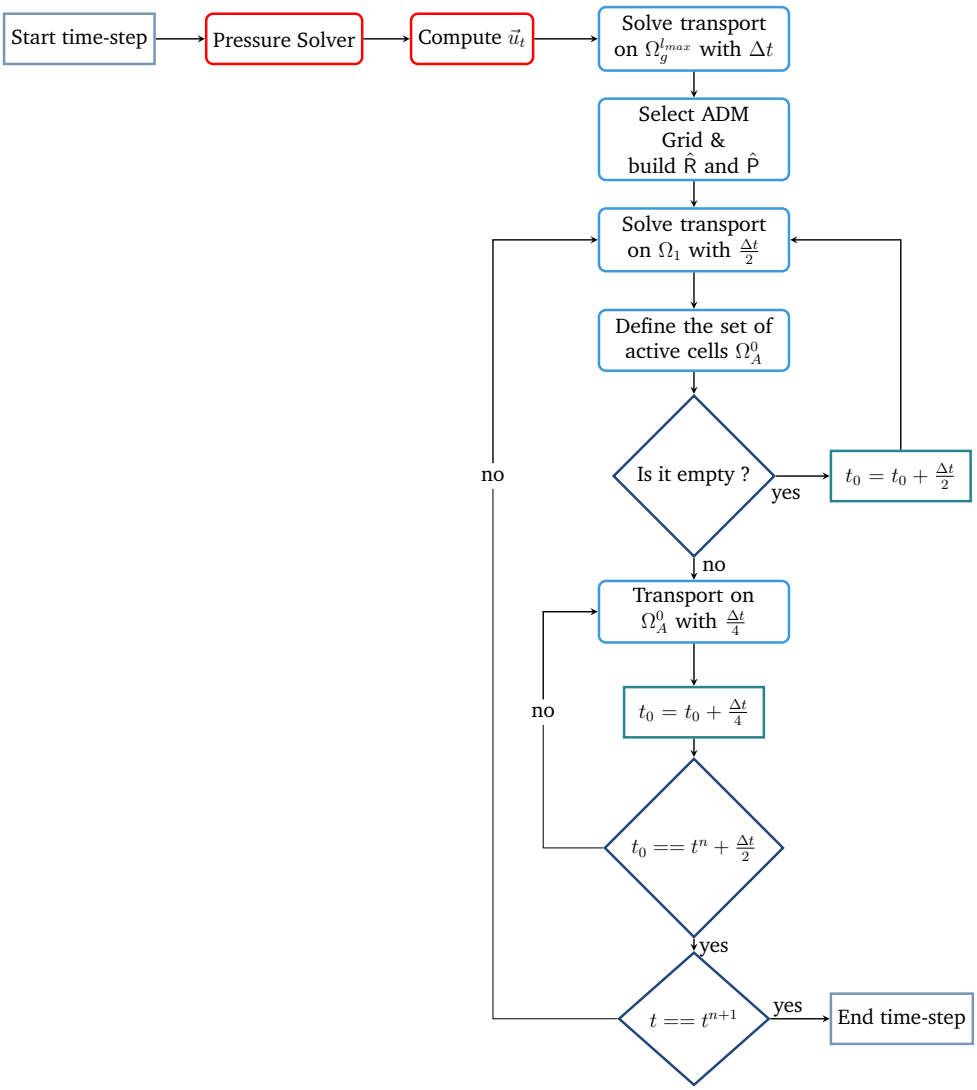


**Fig. 5.1.:** Schematic overview of a time step for the ADM-LTS strategy with $\eta = 2$ and $l_{max} = 2$.

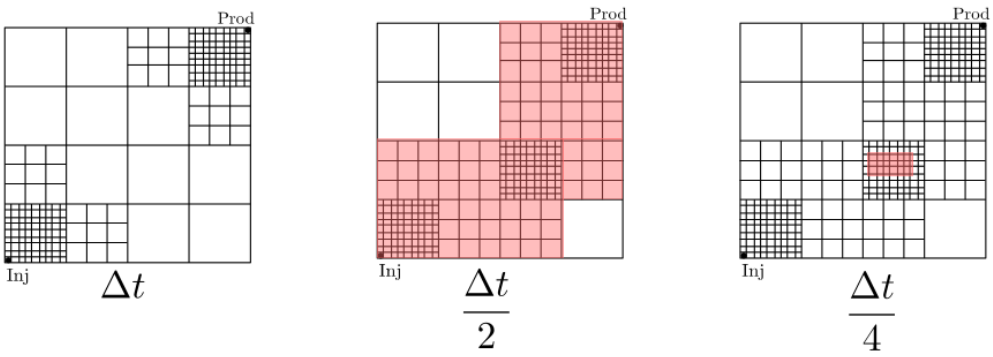


**Fig. 5.2.:** Example of ADM grid and active regions for the refinement time steps with $\eta = 2$ and $l_{max} = 2$.

When solving for the cells belonging to $\Omega^l$, the numerical flux at the interface between two cells $K$ and $L$ such that $K \in \Omega^l \wedge L \in \Gamma^{l+1}$ is approximated by

$$F_{KL}^{n+\beta(i)} = \frac{F_{KL}^{n+\gamma(j)}}{\eta}$$

where

$$\beta(i) = \frac{i}{\eta^{(l_{max}-l)}}, \quad i = \{1 \cdots \eta^{(l_{max}-l)}\}$$

$$\gamma(j) = \frac{j}{\eta^{(l_{max}-l-1)}}, \quad j = \{1 \cdots \eta^{(l_{max}-l-1)}\}.$$

Thus, Eq. (3.14) can be modified to account for the presence of different time levels as follows

$$\begin{aligned} \mathsf{r}_K^{n+\beta(i)} &= \phi(S_K^{n+\beta(i)} - S_K^{n+\beta(i)-1}) \\ &- \frac{1}{|K|} \sum_{e_{KL} \in \mathcal{E}_{KA}} F_{e_{KL}}^{n+\beta(i)} - q_t^{n+\beta(i)} - \frac{1}{|K|} \sum_{e_{KL} \in \mathcal{E}_{KL}} \frac{F_{e_{KL}}^{n+\gamma(j)}}{\eta} \end{aligned}$$

Here, $\mathcal{E}_{KA}$ is the set of interface fluxes exchanged between two cells $K$ and $L$ both belonging to $\Omega^l$. Additionally, $\mathcal{E}_{KL}$ is the set of fluxes at the interface between two cells $K$ and $L$ where $K \in \Omega^l$ and $L \in \Gamma^{l+1}$. Note that, for $l = 0$ the residual for the active cells is the same described by equation (3.14) but $\mathcal{E}_{KL}$ would be the set of fluxes at the interface between $\Omega_A^0$ and $\Omega^1 \setminus \Omega_A^0$ .

Remark that, for each level $l$, the linear system that has to be solved has the size $N_A^l \times N_A^l$ which is significantly smaller than the full fine-scale system. Therefore, the dimensions of $\mathsf{J}$ and $\mathsf{r}$ in (5.1) are smaller. The restriction operator $\hat{\mathsf{R}} = \hat{\mathsf{R}}_l^{l-1} \ldots \hat{\mathsf{R}}_1^f$ and the prolongation operator $\hat{\mathsf{P}} = \hat{\mathsf{P}}_f^1 \ldots \hat{\mathsf{P}}_{l-1}^l$ have to be resized in order to solve the smaller linear system. At the beginning of the time refining strategy, the size of $\hat{\mathsf{R}}$ is equal to $N_{l_{max}}^{ADM} \times N_f$. To obtain the correct restriction matrix ($\hat{\mathsf{R}}_{l_{ref}}$), from the fine grid to the active cell of the ADM grid, we need to select only the rows that represent the active cells; the columns with all zeros entries need to be eliminated. The reduced prolongation operator $\hat{\mathsf{P}}_{l_{ref}}$ is defined as $\hat{\mathsf{P}}_{l_{ref}} = \hat{\mathsf{R}}_{l_{ref}}^T$. So inside each local time step the following system is constructed:

$$\hat{\mathsf{R}}_{l_{ref}} \mathsf{J}_{l_{ref}} \hat{\mathsf{P}}_{l_{ref}} \delta \mathsf{x}_{l_{ref}}^{ADM} = -\hat{\mathsf{R}}_{l_{ref}} \mathsf{r}_{l_{ref}}$$

where $\delta \mathsf{x}_{l_{ref}}^{ADM}$ is the ADM solution of the active adaptive multilevel grid cells. The fine solution of the active grid cells is approximated by

$$\delta \mathsf{x}_{w_A} \approx \hat{\mathsf{P}}_{l_{ref}} \delta \mathsf{x}_{l_{ref}}^{ADM}$$

where the subscripts $A$ represents the subset of active cells on the finest level. The above strategy allows for conservative multiscale march in time and space for transport equation within the sequentially implicit simulation framework. In the following sections, its performance are studied for various test cases both for 2D and 3D domains.

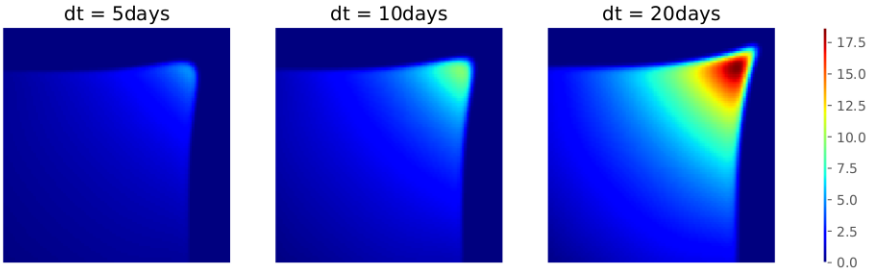


**Fig. 5.3.:** Test case 1 $[99 \times 99]$ – CFL values for different global time steps at time $t = 500$ days.

## 5.3 Numerical results

The performance of the newly developed ADM-LTS strategy is thoroughly investigated for several challenging test cases. For all the cases presented, quadratic relative permeability curves are considered. Additionally, all errors are computed with respect to a reference solution, obtained by employing a high resolution discretization both in space and in time. In all the test cases the gravity ans capillary forces are neglected, except for case test 2 where the gravity forces are introduced, and case test 7 where we analyzed the same scenario with and without considering the capillary effects.

### 5.3.1 Test case 1: 2D homogeneous reservoir

The first test case is a $100 \times 100$ $[\text{m}^2]$ homogeneous reservoir, with isotropic permeability of $5 \times 10^{-15}$ $[\text{m}^2]$. A pressure-constrained wetting-phase injector well is positioned in the bottom-left corner of the domain with a pressure $p_{inj} = 10^8$ [Pa], whereas a production well is present in the top-right corner with a relative pressure of $p_{prod} = 0$ [Pa]. The phase viscosity values are $\mu_w = 10^{-3}$ [Pa $\cdot$ s] and $\mu_{nw} = 10^{-2}$ [Pa $\cdot$ s] for the wetting and non wetting phase, respectively. The final simulated time is $600$ [days] after injection has started.

A fine-scale grid with $99 \times 99$ cells is imposed on the domain. ADM-LTS employs a time refining ratio $\eta = 2$ and a space coarsening ratio equal to 3 in all directions. The user-defined tolerances for the coarsening and refinement criteria are $\epsilon_x = 0.07$ and $\epsilon_t = 5 \times 10^{-2}$.

Simulations are run employing three different global time step sizes: 5, 10 and 20 days. Fig. 5.3 reports the CFL values at time t = 500 days for the three different time steps for fine-scale in space simulations.

Fig. 5.4 shows a comparison of the ADM-LTS solution with the reference solution at time t = 500 [days] using three different sizes of the global time steps, the space grid obtained at the final time are similar capturing the sharp front.

Fig. 5.5 reports the error for the saturation at time t = 500 days between a reference solution and the ADM method with fine time steps (first column) with the LTS approach (second column) and with the coarse time steps (third column) for the three different time steps sizes $\Delta t = 5$ (first row), $\Delta t = 10$ (second row) and $\Delta t = 20$ (third row). In all cases the AMD-LTS approach

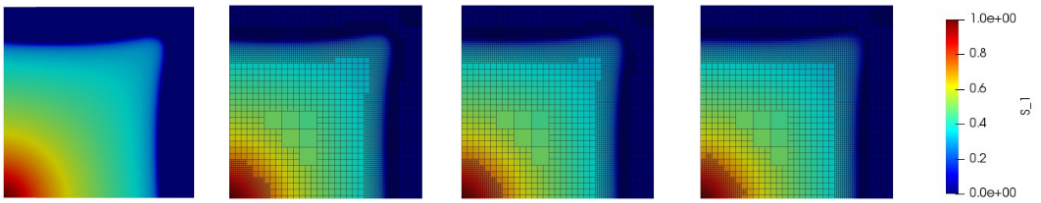


**Fig. 5.4.:** Test case 1 $[99 \times 99]$ – Reference solution (first column) and ADM-LTS solution using a global time step size: $\Delta t = 5, 10$ and $20$ [days] at time t = 500 days for the second, third and fourth column, respectively.

improves the errors with respect to the reference solution of the coarse time step approach.

The complexity of the algorithm is shown in Fig. 5.6. In particular, each column represents the total amount of active cells multiplied by the number of Newton iterations involved to compute the solution, for the three approaches and for the three different global time step sizes. Note that, to obtain the solution at time $t = 600$ [days], $120, 60$ and $30$ global time steps have been performed using the three analyzed time steps. We remark that the errors obtained by employing the original ADM method with a fine time step are comparable to those obtained with ADM-LTS in terms of accuracy.

Fig. 5.7 shows the complexity of a single global time step. For the ADM method with fine time steps, the local steps are just the small steps applied at the whole domain. At the end of the local steps both the ADM-LTS method and the ADM fine step method reach the same time. For the ADM-LTS method, local step 1 indicates the global step on the coarsest grid, step 2 and 5 are the intermediate time steps performed on level 0 and 1 of the ADM grid, and the other local steps are the small time steps for the active cells detected by the error estimator in time. In particular, we can notice that the intermediate time steps have almost the same complexity of the small time steps of the ADM fine method, even if the size of the time step is two times bigger with almost the same number of active cells. This is due to the improvement of the initial guess for the Newton loop. In the intermediate time steps we use as initial guess a linear combination of the solution of the previous time $t^n$ and the solution obtained on the coarsest grid at the new global time $t^{n+1}$. In the small time steps is not necessary to perform this technique since a small step is used to advance in time.

The same test case is analyzed after performing a $2 \times 2$ refinement of the space fine-scale grid. In order to obtain a reasonable solution, using a global $\Delta t$ equal to 20 [days], we need to compute more local time steps inside the global one, so a refining ratio equal to 4 has been taken into account.

Fig. 5.8 reports the complexity for the entire simulation using ADM-LTS method and the ADM with fine time steps. To obtain the solution at final time $t = 600$ [days] with a global time step equal to 20 days, the same number of global time step are involved (30 time steps in total). Of course the number of local time step for both the LTS method and the fine time steps approach has increased; but the ratio between active cells and total cells decreases.

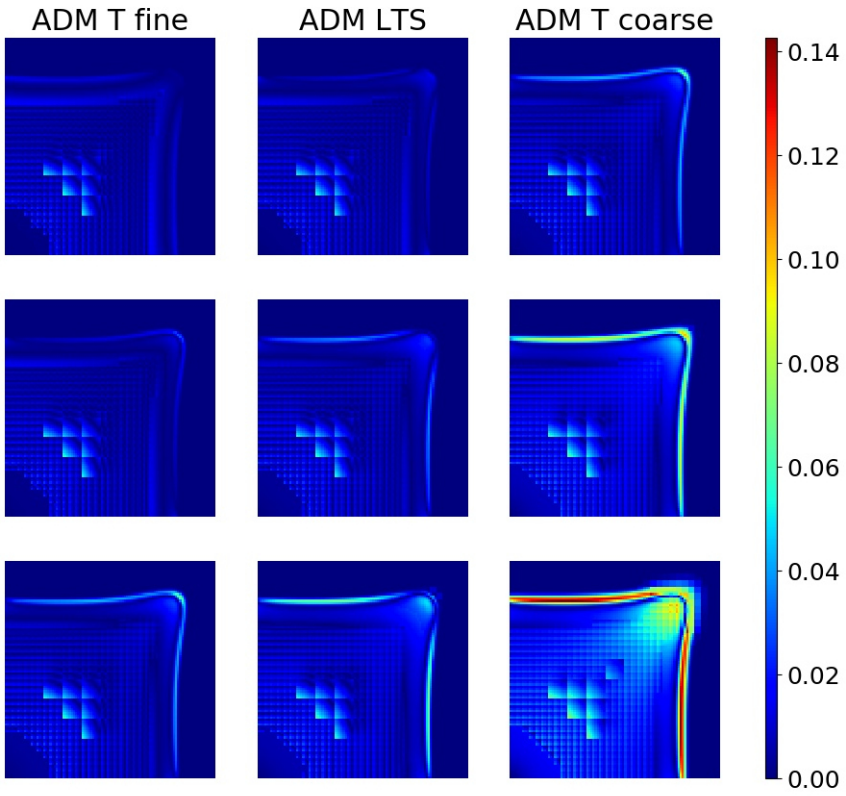


**Fig. 5.5.:** Test case 1 $[99 \times 99]$ – Saturation errors for the ADM method with fine time steps (first column), ADM-LTS method (second column) and ADM coarse time steps method (third column) for the three different global time step sizes.

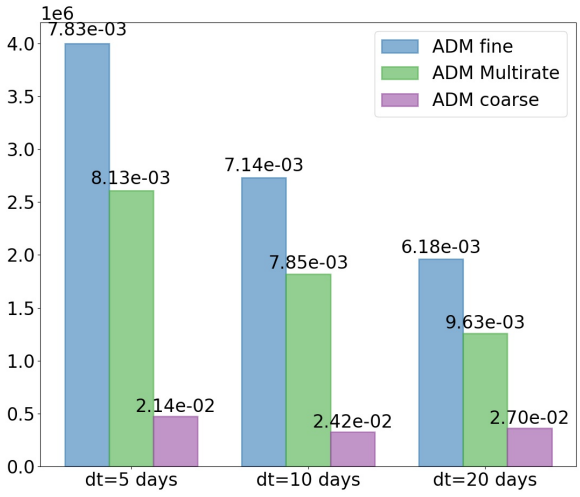


**Fig. 5.6.:** Test case 1 $[99 \times 99]$ – Total amount of active cells multiplied by number of Newton iterations for the three different time step sizes. On the top of each bar the mean in time of the averaged absolute difference respect to the reference solution for the saturation is displayed $E_s = \text{mean}|S(t_f) - S_{ref}(t_f)|$ where $t_f$ is the final time 600 days.

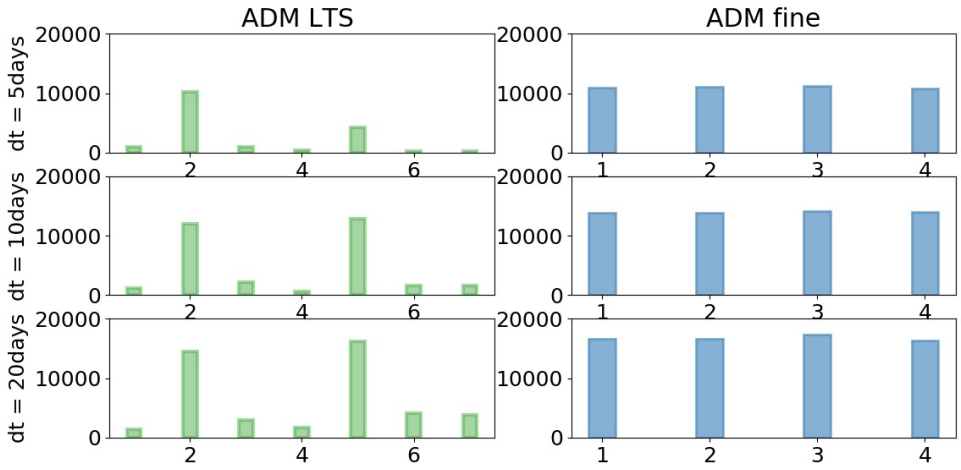


**Fig. 5.7.:** Test case 1 $[99 \times 99]$ – Computational complexity history at each local times step within a global step. The computation complexity is the number of active cells multiplied by the number of Newton iterations.

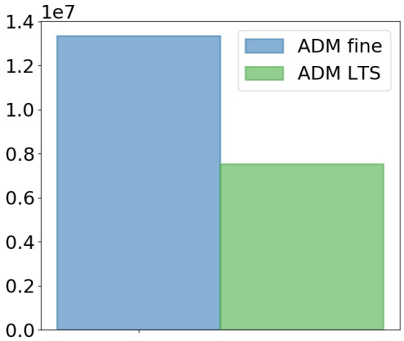


**Fig. 5.8.:** Test case 1 $[198 \times 198]$ – Total amount of active cells multiplied by number of Newton iterations for the ADM with fine time steps and the ADM LTS method.

Fig. 5.9 shows the averaged number of active cells times the number of Newton iterations for each local time step within a global time step.

In Fig. 5.10 we can see that the ADM-LTS approach reduces the errors obtained using a coarse grid in time.

### 5.3.2 Test case 2: 3D homogeneous reservoir

A 3D $108 \times 108 \times 108$ $[\mathrm{m}^3]$ homogeneous reservoir is considered in this test case. The domain is discretized, at the fine-scale, with a $54 \times 54 \times 54$ Cartesian grid for a total of $157464$ cells. The physical parameters are the same of the first test case. The size of the global steps is equal to 125 days. The simulation ends after 70 global steps. The tolerances for the coarsening criteria in space and time are set to be $\epsilon_x = 0.2$ and $\epsilon_t = 5 \times 10^{-2}$.

Fig. 5.11 reports the saturation maps at two different simulation times (on the top) without considering the gravity effects. The set of active cells $\Omega_A^0$ at time $t = 1500$ (left, bottom) and also a section of the solution at final

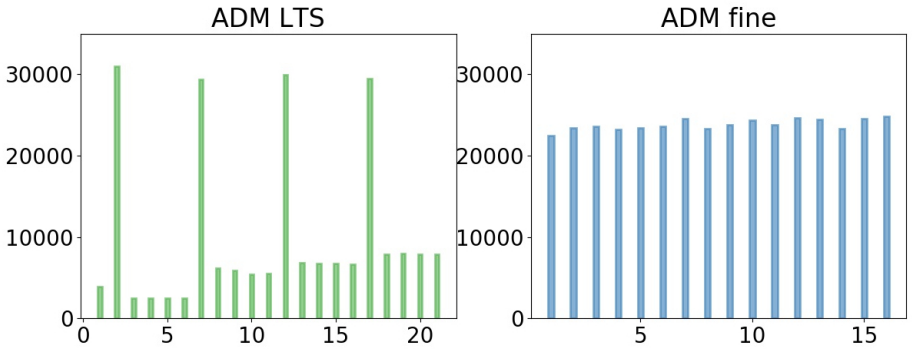


**Fig. 5.9.:** Test case 1 $[198 \times 198]$ – Computational complexity history at each local times step within a global step. The computation complexity is the number of active cells multiplied by the number of Newton iterations.

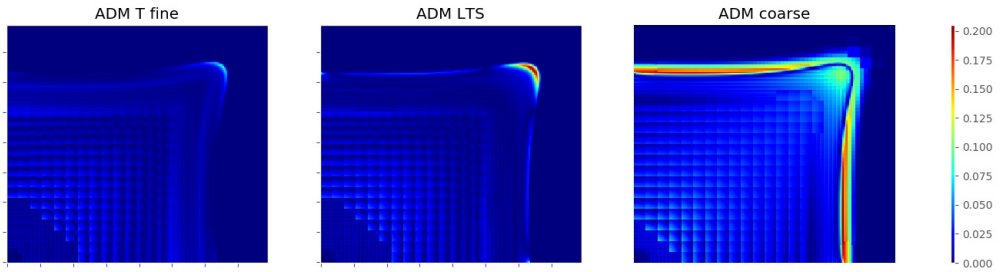


**Fig. 5.10.:** Test case 1 $[198 \times 198]$ – Saturation errors at time t = 540 [days] for the ADM method with fine grid in time (left), ADM-LTS method (center) and the ADM method with coarse grid in time (right).

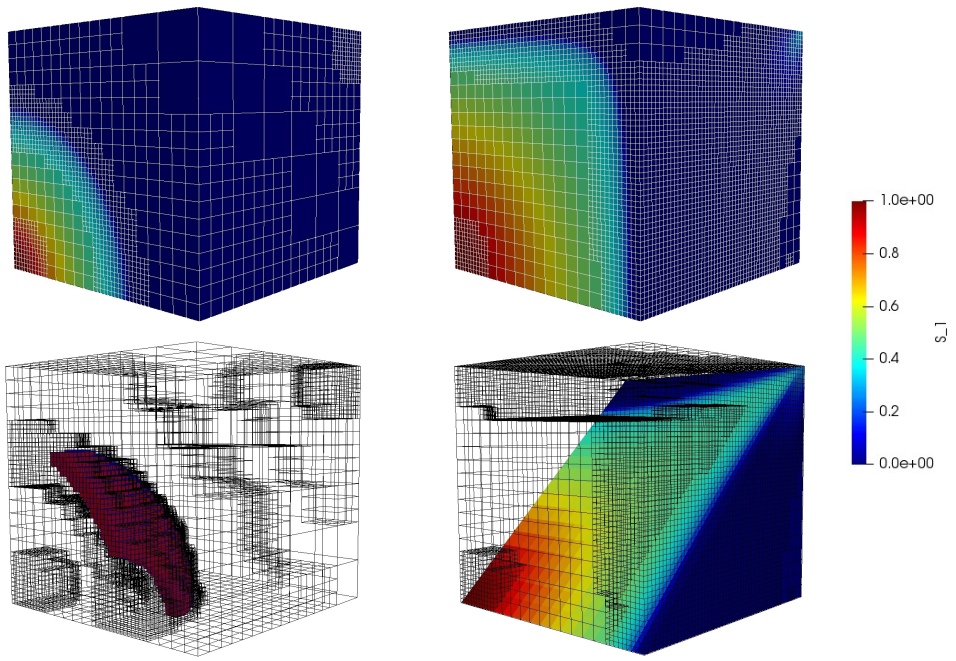


**Fig. 5.11.:** Test case 2 – Saturation profile (top row) at time t = 1500 days (left) and at time t = 8750 days (right). Active cells for the level $l_{ref=0}$ at time t= 1500 (bottom-left) and saturation profile inside the domain at time $t = 8750$ (bottom-right).

time t = 8750 days (right, bottom) are also plotted. Note that ADM-LTS automatically employs fine cells only around the advancing saturation front and that the active cells in time are only a fraction of them. Fig. 5.14 shows the total complexity and the mean complexity per local time step for both the ADM-LTS method and the ADM method with fine time steps. The ADM-LTS results cheaper with respect to the classical ADM method. The complexity ratio is equal to $0.6$.

Fig. 5.13 shows the saturation maps and the active cells $\Omega_A^0$ at different times considering the same scenario but introducing the buoyancy forces. The density values of the wetting and non-wetting phase have been changed, the ratio is equal to $\rho_w/\rho_{nw} = 5/4$. As expected, the wetting profile is not radially symmetric for the presence of the gravity forces that bring the fluids at the bottom of the reservoir.

Fig. 5.14 shows the total complexity and the mean complexity per local time step for both the ADM-LTS method and the ADM method with fine time steps, with and without considering the gravity effects. The transport equation is highly nonlinear in the presence of buoyancy forces, so the nonlinear loops need a large number of Newton iterations to converge for both the classical ADM method and the new ADM-LTS approach. Instead, as shown in Fig. 5.15, the number of active cells employed by the ADM-LTS approach for the two different scenarios is the same for the entire simulation.

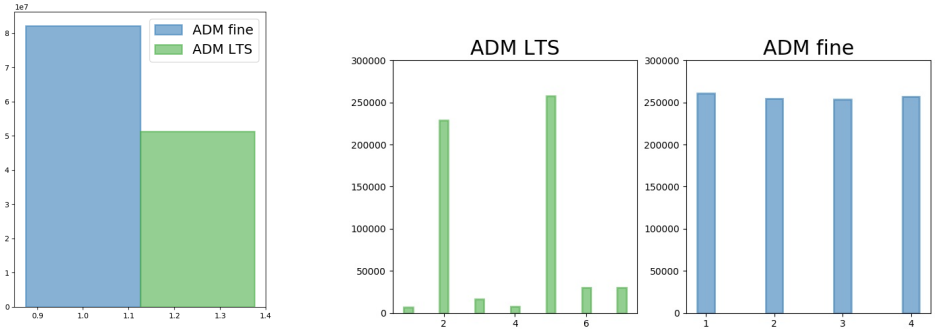


**Fig. 5.12.:** Test case 2 – Total amount of active cells multiplied by number of Newton iterations for the ADM with fine time steps and the ADM LTS method (left) and computational complexity history at each local times step within a global step (right).

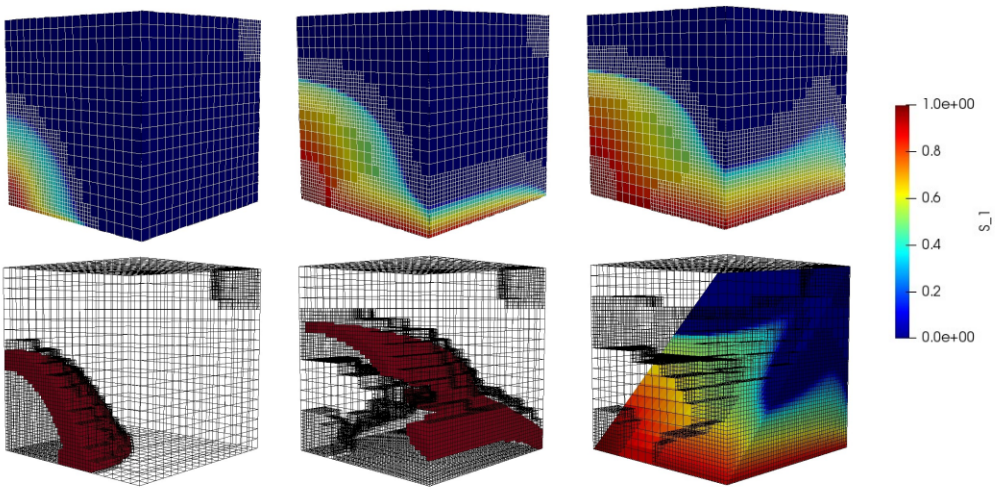


**Fig. 5.13.:** Test case 2 - Saturation profile (top row) at time t = 1500 days (left), t = 6375 (center) and t = 8750 days (right). Active cells for the level $l_{ref=0}$ at time t= 1500 (bottom-left), active cells for the level $l_{ref=0}$ at time t= 6375 (bottom-center) and saturation profile inside the domain at time $t = 8750$ (bottom-right).

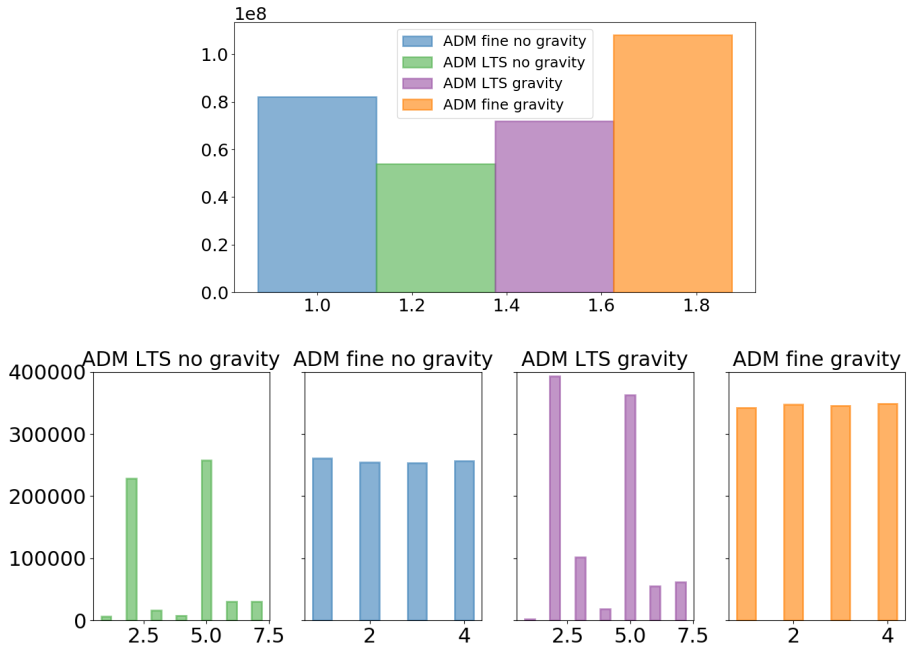


**Fig. 5.14.:** Test case 2 – Total amount of active cells multiplied by number of Newton iterations for the ADM with fine time steps and the ADM LTS method (left) and computational complexity history at each local times step within a global step (right) with and without buoyancy forces.

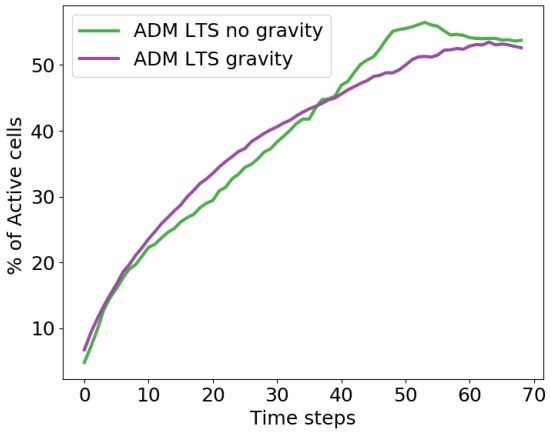


**Fig. 5.15.:** Test case 2 – Number of active cells employed in the ADM-LTS simulations with and without gravity effects.

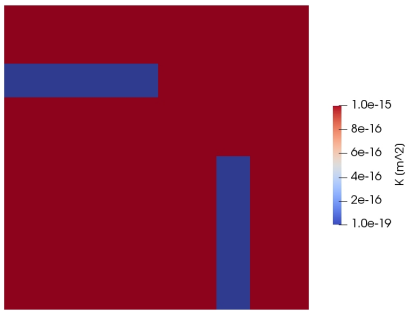


**Fig. 5.16.:** Test case 3 – Absolute permeability field.

### 5.3.3 Test case 3: 2D homogeneous reservoir with barrier

A 2D homogeneous reservoir with low permeability barriers is considered, as shown in Fig. 5.16. The same permeability field was presented in [22]. The domain dimensions and the physical parameters are the same of the first test case, the same $99 \times 99$ fine scale grid is imposed. The global time step is equal to $50$ [days] and the simulation ends after $100$ global time steps ($t = 5000$ days).

Simulations are carried out both with the original ADM method employing a global fine time-step and with ADM-LTS. The coarsening and the time-refinement criteria tolerances are set to $\epsilon_x = 0.05$ and $\epsilon_t = 0.005$.

Fig. 5.17 shows a comparison of the saturation profile and the grid resolution for the two different strategies. The original ADM method with a saturation difference-based coarsening criterion (top row) employs a large number of fine grid cells wherever saturation gradients are present even if they are stationary. On the other hand the newly proposed grid resolution criterion (bottom row) for the ADM-LTS approach uses fine cells only in those regions where the saturation gradient is moving, reducing the number of active cells.

Fig. 5.18 reports the evolution of the active grid cells percentage for the two different approaches (left) and the evolution of the relative saturation error in $l^1$-norm (right). In the early steps, we can see that the ADM fine with just the gradient criterion approach employs almost the same number of active grid cells used by the ADM-LTS method. For the ADM with fine time steps at every small local time step we solve both the flow and the transport equations, instead for the ADM-LTS approach only the transport equation is solved for the local steps. This is the reason why in the first five steps the saturation errors for the ADM-LTS approach are larger with respect to the ADM fine steps approach. Instead, in the last steps the errors increase because a lower number of fine grid cells has been used.

Fig. 5.19 shows the total complexity (number of active cells multiplied by the number of Newton iterations) for the ADM with fine steps and the

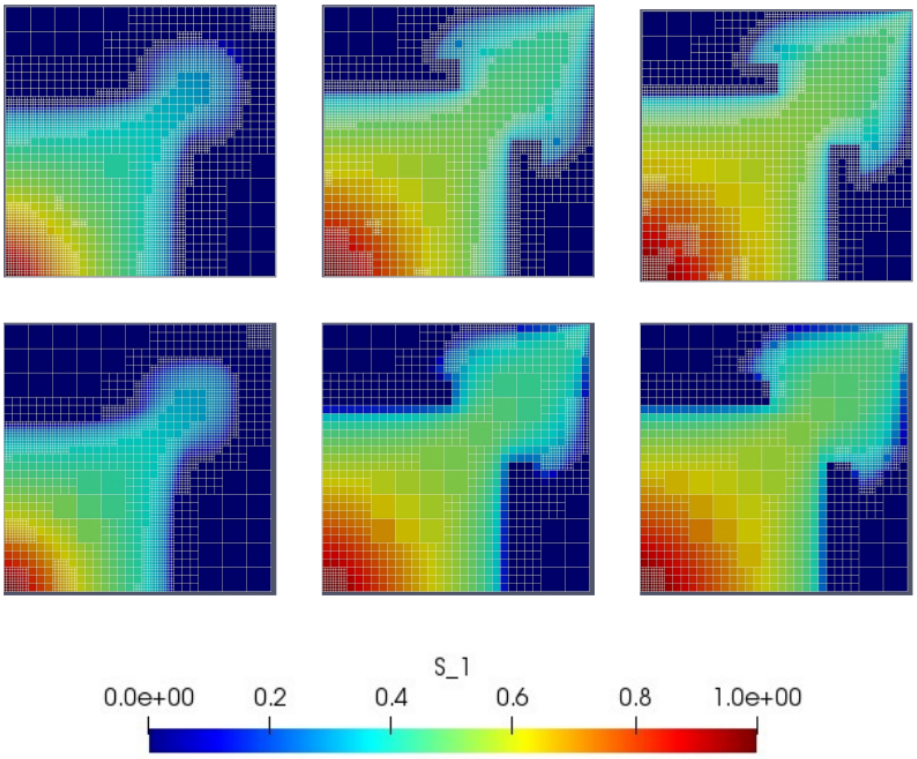


**Fig. 5.17.:** Test case 3 – Saturation profile and ADM grid at different time steps (columns) for ADM with coarse time steps and classical ADM grid resolution (first row) and for ADM-LTS method with the new ADM grid resolution (second row).

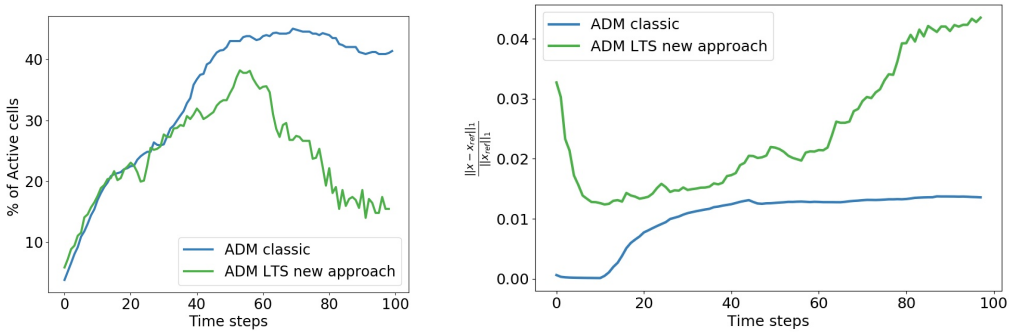


**Fig. 5.18.:** Test case 3 – number of active cells employed in ADM with fine grid in time and ADM-LTS simulations expressed as percentage of fine grid cells (left) and the saturation relative errors in $l^1$-norm for the ADM fine and ADM-LTS method (right).

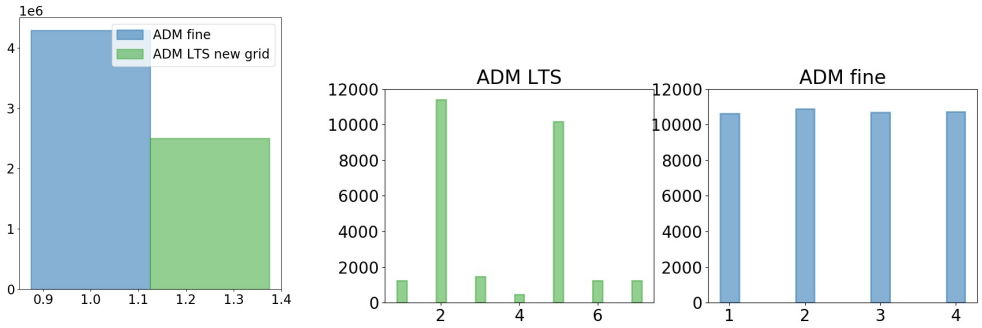


**Fig. 5.19.:** Test case 3 – Total amount of active cells multiplied by number of Newton iterations (left) and computational complexity history at each local times step within a global step (right) for the ADM with fine time steps approach and for the ADM-LTS method.

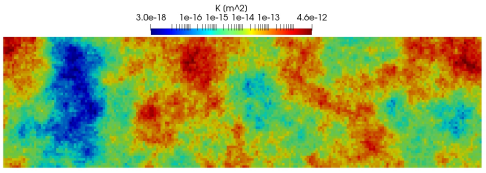

**Fig. 5.20.:** Test case 4 – Natural logarithm of the permeability.

ADM-LTS approach. Note that the local time steps of ADM-LTS method reduce the complexity of the system compared to the classical ADM approach.

### 5.3.4 Test case 4: Heterogeneous reservoir (SPE10 top layer)

In this test case a heterogeneous reservoir is considered. The permeability map is the top layer of the SPE10 test case [17] and it is presented, in logarithmic scale, in Fig. 5.20. The size of the reservoir is $2200 \times 600$ $[\mathrm{m}^2]$ and a $216 \times 54$ grid is employed at the finest level. The injector is at the top left corner and has a constrained pressure $10^7$ [Pa]. A producer is, instead, located at the bottom right corner of the domain with a pressure equal to 0 [Pa]. The porosity of the reservoir $\phi$ is equal to 0.2. The viscosity for wetting phase is $10 \times 10^{-5}$ [Pa $\cdot$ s], whereas, for the non-wetting phase, is $10^{-4}$ [Pa $\cdot$ s]. The coarsening ratio for the space grid is equal to 2 as well as the time refining ratio. The error tolerance for the time estimator is equal to $5 \times 10^{-2}$.

Fig. 5.21 reports the saturation map and the ADM grid for different threshold values of the ADM grid resolution criterion using the classical ADM approach with fine time steps, and the ADM-LTS approach with the new grid resolution strategy. The classical approach uses, for small threshold values, a large number of fine grid cells. If we relax the threshold parameter the method is not able to capture the fronts. Thanks to the new ADM-LTS approach, the method is able to apply the fine grid cells only where the front is moving fast (high permeability regions).

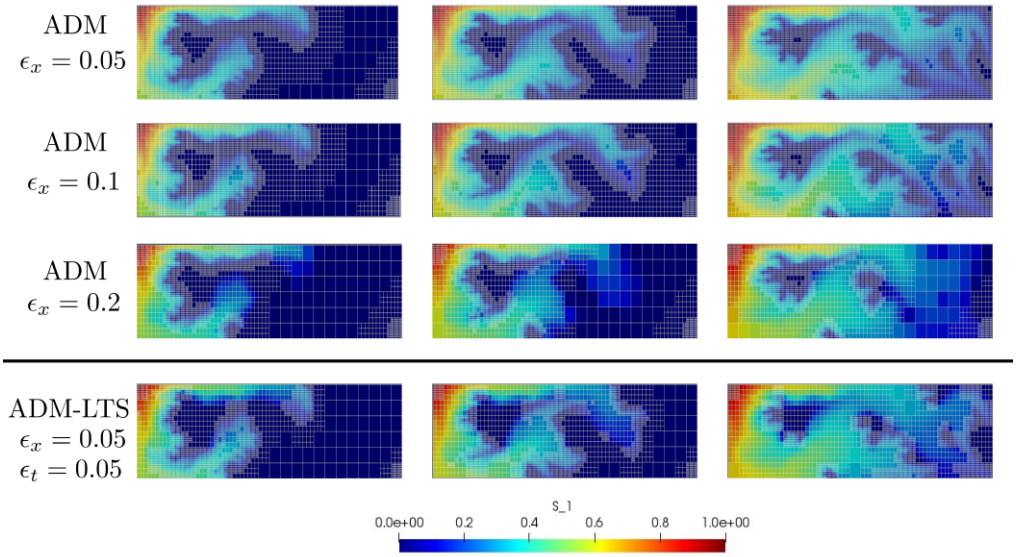


**Fig. 5.21.:** Test case 4 – Saturation map and ADM grid for the ADM with fine time step approach with classical grid criterion for different values of the threshold $\epsilon_x = 0.05, 0.1, 0.2$ (row 1, 2, and 3) and for the ADM-LTS method with the new grid criterion $\epsilon_x = 0.05$ and $\epsilon_t = 0.05$ (row 4) at time $t = 1200$ days (first column), $t = 15000$ days (second column) and $t = 20000$ days (third column).

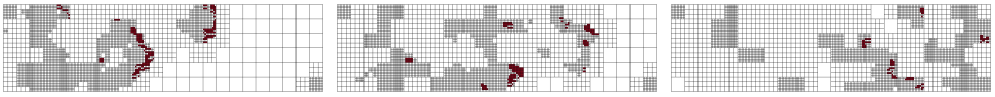

**Fig. 5.22.:** Test case 4 – Active cells for the refinement in time, at time $t = 1200$ days (left), $t = 15000$ days (center) and $t = 20000$ days (right).

Fig. 5.22 shows the active cells in time at the finest level $l_{ref} = 2$ for different global time steps. The method recomputes the solution with small time steps only for a few percentage of cells where the front crosses high permeability regions. In fact, in the last snapshot, the saturation profile is completely developed everywhere and so, the set of active cells is very small.

In Fig. 5.23 we compare the number of active cells and the saturation errors for the different simulations. Using the classical ADM approach with small values of the tolerance a lot of active grid cells are employed giving very small errors. The classical ADM approach with larger tolerance value and the ADM-LTS method are comparable in terms of active cells during all the simulation but the ADM-LTS approach gives better results in term of errors.

Fig. 5.24 reports the complexity of the four simulations. The ADM approach with fine grid in time and small threshold values is really expen-

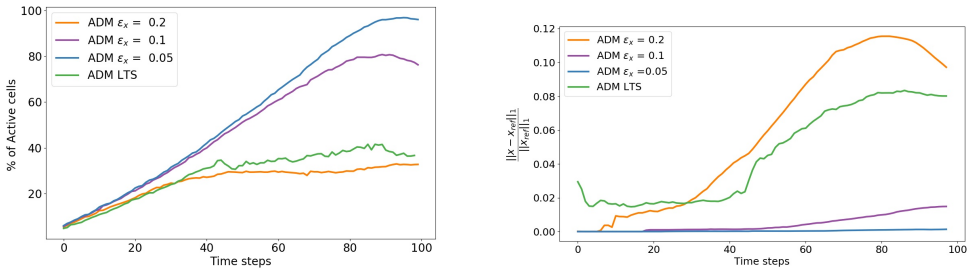


**Fig. 5.23.:** Test case 4 – Number of active cells expressed as percentage of fine grid cells (left) and saturation relative errors in $l^1$-norm (right) for the ADM with fine grid in time with different values of the threshold and for the ADM-LTS simulation.

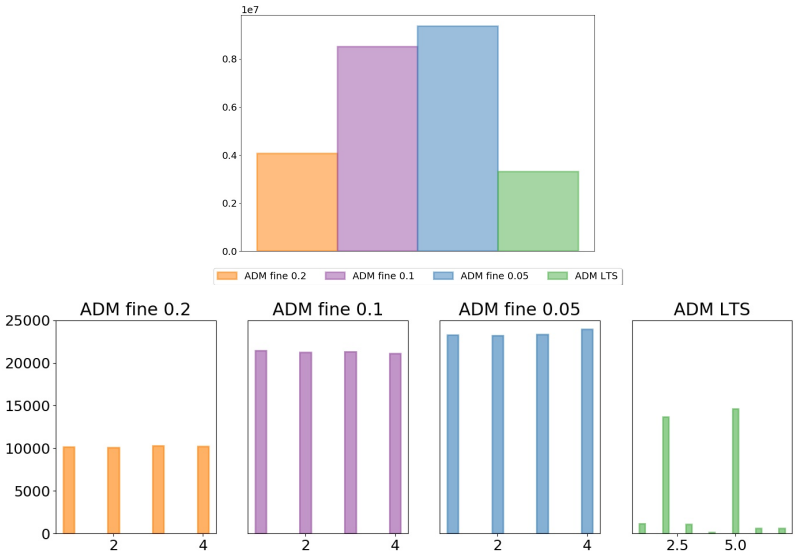


**Fig. 5.24.:** Test case 4 – Total amount of active cells multiplied by number of Newton iterations (top) and computational complexity history at each local times step within a global step (bottom) for the ADM approach.

sive. The ADM-LTS approach is comparable to the ADM with fine time step approach and large value of $\epsilon_x$ but, as shown previously, the solution of the classic ADM, in this case, is not as accurate.

### 5.3.5 Test case 5: Heterogeneous reservoir (SPE10 bottom layer)

The permeability of SPE10 bottom layer is used for this test case, as show in Fig. 5.25. This layer, with respect to the previous case has higher contrasts and more a channelized distribution.

The global time step is equal to 10 days and the simulation ends after 50 global time steps. The input parameters for the wells and the physical properties are identical to Test Case 4.

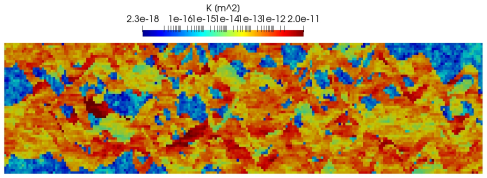


**Fig. 5.25.:** Test case 5 – Natural logarithm of the permeability.

The top 2 rows of Fig. 5.26 show the saturation distribution at simulation time of $150$, $250$ and $350$ days obtained with $\epsilon_x = 0.15$ and $\epsilon_x = 0.2$, respectively. The bottom rows, instead, show the saturation map, at the same simulations times, obtained by employing the ADM-LTS method with $\epsilon_x = 0.05$ and $\epsilon_t = 5 \times 10^{-2}$ and $\epsilon_t = 5 \times 10^{-3}$. The classic ADM approach employs a large number of active cell during the simulation, instead the ADM-LTS method is able to select a fine scale grid only where the fronts are moving.

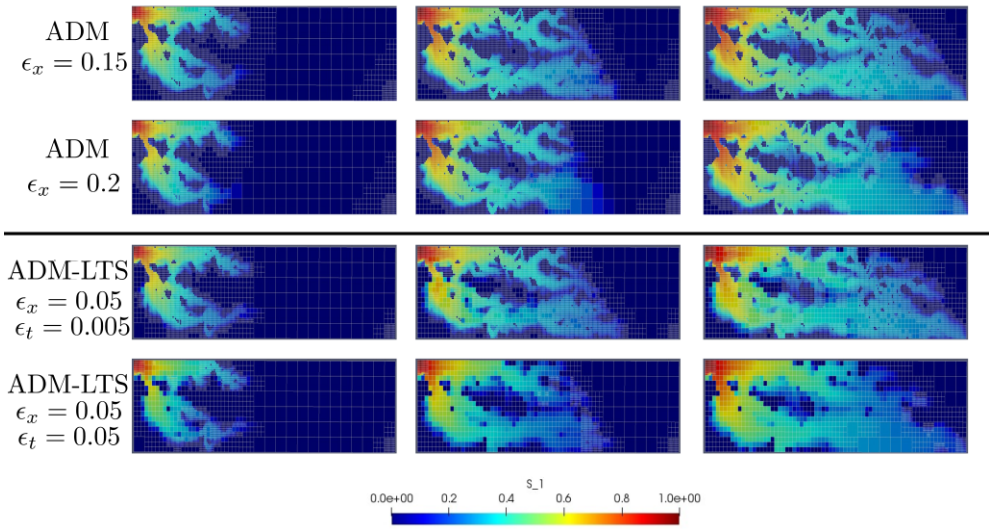


**Fig. 5.26.:** Test case 5 – Saturation map and ADM grid at $150$, $250$ and $350$ days for the ADM approach with fine time steps and the ADM-LTS approach.

Fig. 5.27 shows the active cells for $l_{ref} = 2$ at time $150$, $250$ and $350$ days. As expected, for smaller value of the threshold more cells are involved in the refinement step.

The history of the percentage of active cells employed by the different simulation strategies for the various tolerances is shown in Fig. 5.28 (left), along with the $l^1$ norm of the saturation error (right). For both the ADM-LTS tolerance values less active cells are involved respect to the classical ADM approach. Since a smaller number of cells is employed, the saturation errors are higher but still of the same order of magnitude.

Fig. 5.29 reports the complexity of the four simulations for different tolerance values.

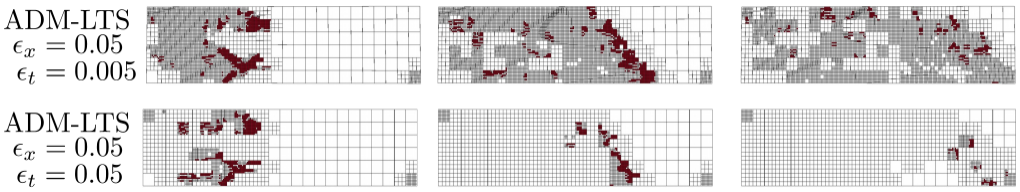


**Fig. 5.27.:** Test case 5 – Active cells for the refinement level $l_{ref} = 2$, at $150$ (left), $250$ (center) and $350$ (right) days for the two threshold values.

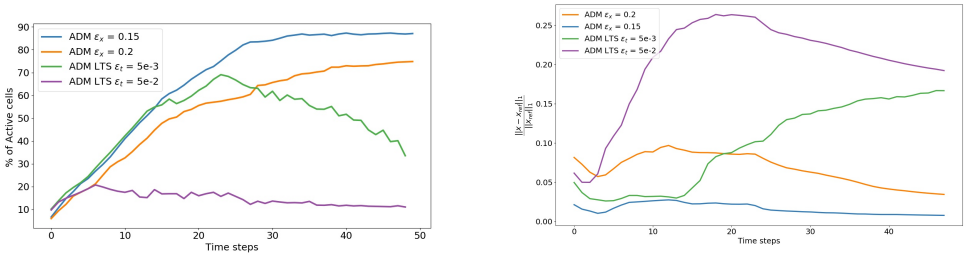


**Fig. 5.28.:** Test case 5 – Number of active cells expressed as percentage of fine grid cells (left) and saturation relative errors in $l^1$-norm (right) for the ADM with fine grid in time and for the ADM-LTS simulations.

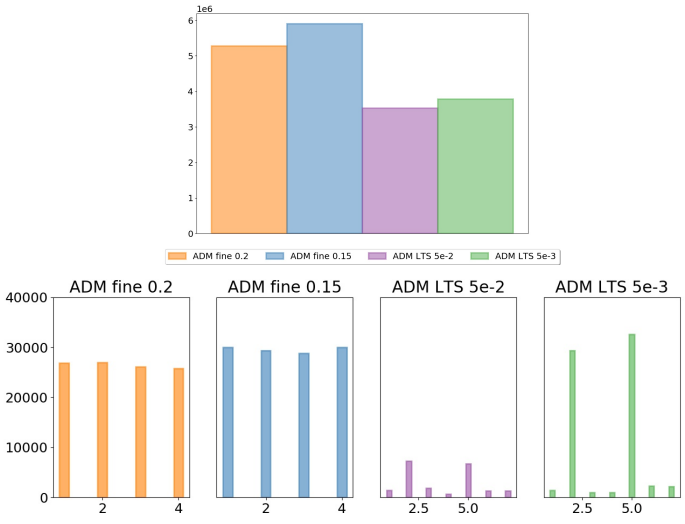


**Fig. 5.29.:** Test case 5 – Total amount of active cells multiplied by number of Newton iterations (top) and computational complexity history at each local times step within a global step (bottom) for the ADM approach.

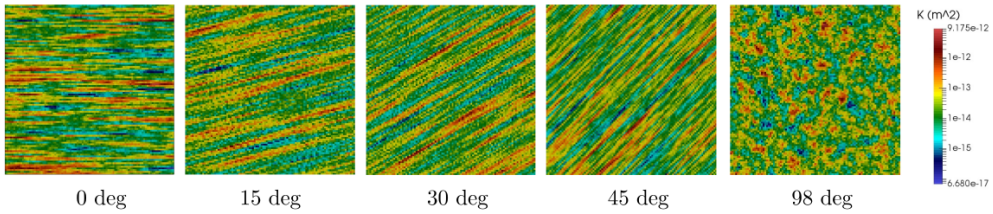


**Fig. 5.30.:** Test case 6 – One of the 20 realization of each of the 5 sets of permeability fields with different angles (0 deg, 15 deg, 30 deg, 45 deg and patchy from left to right).

### 5.3.6 Test case 6: Heterogeneous reservoirs with different layering orientations.

A $500 \times 500$ m$^2$ 2D reservoir is considered on which a $99 \times 99$ grid is imposed. The fluid properties, the location of the wells and their constraints are the same as in the previous test cases. Five sets of permeability fields, with different layering orientation and created using sequential Gaussian simulations with spherical variogram and dimensionless correlation lengths, $0.5$ and $0.02$ as proposed [82], are considered. Each set consists of 20 statistically identical realizations.

Fig. 5.30 shows one realization for each set. Injection of the wetting phase, for 560 days, is simulated for each realization. Simulations are run with the ADM-LTS method. For all runs, the spatial coarsening criterion tolerance is $\epsilon_x = 0.008$. Two different values are instead considered for the time-based criterion tolerance, $\epsilon_t$: $5 \times 10^{-2}$ and $5 \times 10^{-3}$.

Fig. 5.31 shows a comparison, for one permeability realization of each set, of the saturation map at the end of the simulation obtained with fine-scale (time and space) simulation (top row), ADM-LTS employing a fixed refined time-step.

Fig. 5.32 displays the active cells in time for the last refinement level of the last global time step. As expected, using a bigger value of the tolerance for the time error estimator, just few cells need to be computed with small time steps, moreover also the space grid changes and allows to use coarser grid cells.

Fig. 5.33 represents the mean and the standard deviations of the complexity for the ADM-LTS method using the two different time-based criterion tolerances and for the solution computed with the fine grid resolution both in space and in time. Note that the y-axis scale for the two pictures are different.

Fig. 5.34 shows the mean and the standard deviations of the saturation errors respect to the reference solution for the ADM-LTS method using the two different time-based criterion tolerances. From these studies, one can conclude that the ADM-LTS performs robustly when several equiprobable realizations are considered. In other words, the error and computational complexities for all 20 realizations are not much different compared with the average values.

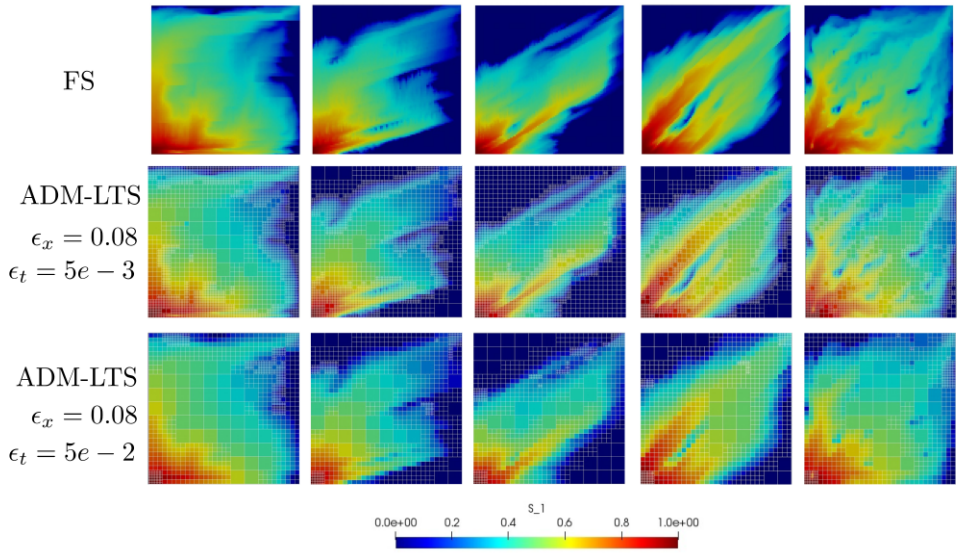


**Fig. 5.31.:** Test case 6 – Comparison of the saturation profile, for one realization of each set of permeability fields at time t = 560 days. Two different threshold values for the time error estimator are employed for the ADM-LTS simulation (center row and bottom row), the fine scale solution are also shown (top row).

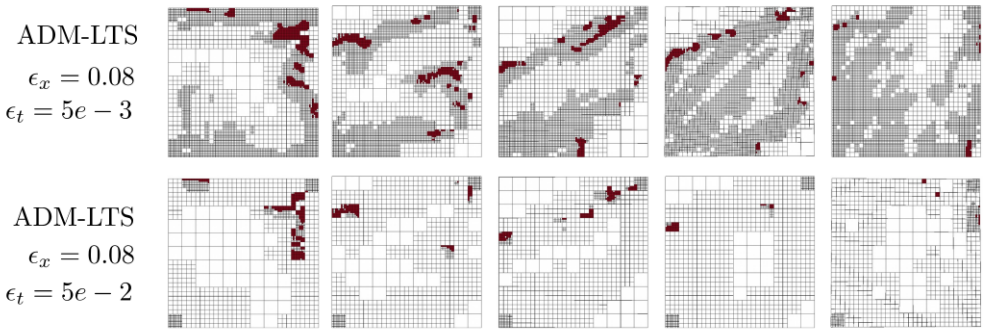


**Fig. 5.32.:** Test case 6 – Active calls at the last refinement level for the last global time step using two different threshold values for the error estimator in time.

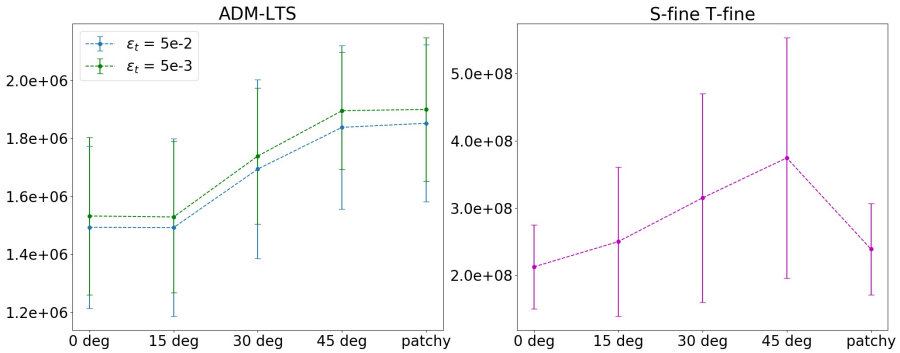


**Fig. 5.33.:** Test case 6 – Mean and standard deviation of complexity over 20 realization for the ADM-LTS method (left) and for the reference solution computed with fine grid resolution both in space and time (right).

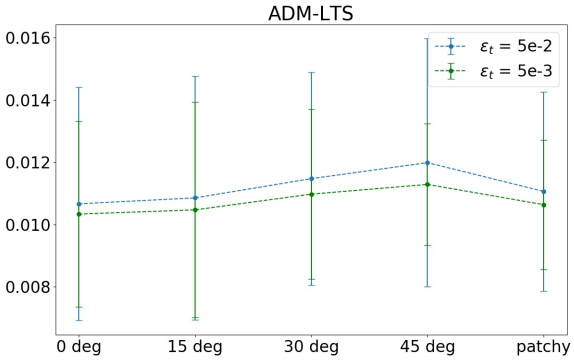


**Fig. 5.34.:** Test case 6 – Mean and standard deviation errors of the saturation errors over 20 realization for the ADM-LTS method with different time threshold values respect to the reference solution $E_S = mean_{t=1}^{N_t}\,(mean|S_f(t) - S(t)|)$.

| Well | x | y | Pressure [bar] |
|---|---|---|---|
| Prod (W1) | 1 | 1 | 120 |
| Prod (W2) | 99 | 1 | 120 |
| Prod (W3) | 99 | 99 | 100 |
| Prod (W4) | 1 | 99 | 100 |
| Inj (W5) | 50 | 50 | 150 |

**Tab. 5.1.:** Test case 7 - Wells coordinates and constrains.

### 5.3.7 Test case 7: Capillary forces.

This test case is the same reported in [24], a $500 \times 500[\ \mathrm{m}^2]$ heterogeneous reservoir is considered, the fine-scale grid contains $99 \times 99$ cells. The phase viscosity are for the wetting and non-wetting phase equal to $\mu_w = 1e-4[\mathrm{Pa}\cdot\mathrm{s}]$ and $\mu_{nw} = 1e-3[\mathrm{Pa}\cdot\mathrm{s}]$, respectively. Five pressure-constrained wells are present, they are showed in Fig. 5.35 as the heterogeneous permeability field. The well locations and the pressure values are presented in Tab. 5.1.

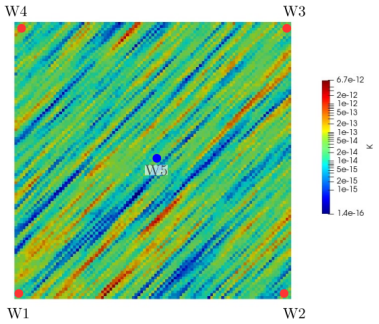


**Fig. 5.35.:** Test case 7: base 10 logarithm of the permeability field.

The final time simulated in $2500[\mathrm{days}]$, with a time step sizes equal to $10[\mathrm{days}]$. The ADM-LTS approach employs a time refining ratio $\eta = 2$ and a space coarsening ratio equal to $3$. The user-defined tolerances are $\epsilon_x = 0.25$ and $\epsilon_t = 1 \times 10^{-2}$. We considered as capillary pressure curve, the J-Leverett Capillary pressure function described in Section 3.2.1, where $\sigma = 4.361 \times 10^{-2}[\mathrm{Pa}\cdot\mathrm{m}]$, $\theta = 0$ and $\gamma = 0.05$. Fig. 5.36 compares the reference solution (fine-scale) and the ADM-LTS approach with and without considering the capillary pressure effects. The ADM-LTS approach, as showed in Fig. 5.37, in presence of capillary pressure involves more fine resolution cells, since the saturation map in more complex with respect to the case without capillary effects. In this small case test the complex behavior is present in the entire domain, so the $70\%$ of active cells are used to capture the heterogeneous capillary pressure, but in real test cases the fine-scale resolution, employed to capture the capillary heterogeneity, usually appear in a small portion of the domain. Fig. 5.37 shows the errors in $l^1$ norm of the ADM-LTS approach with and without capillary capillary pressure. Before the global step 150 the errors for the saturation map with capillary pressure are smaller with respect to the errors without the capillary pressure; instead, at

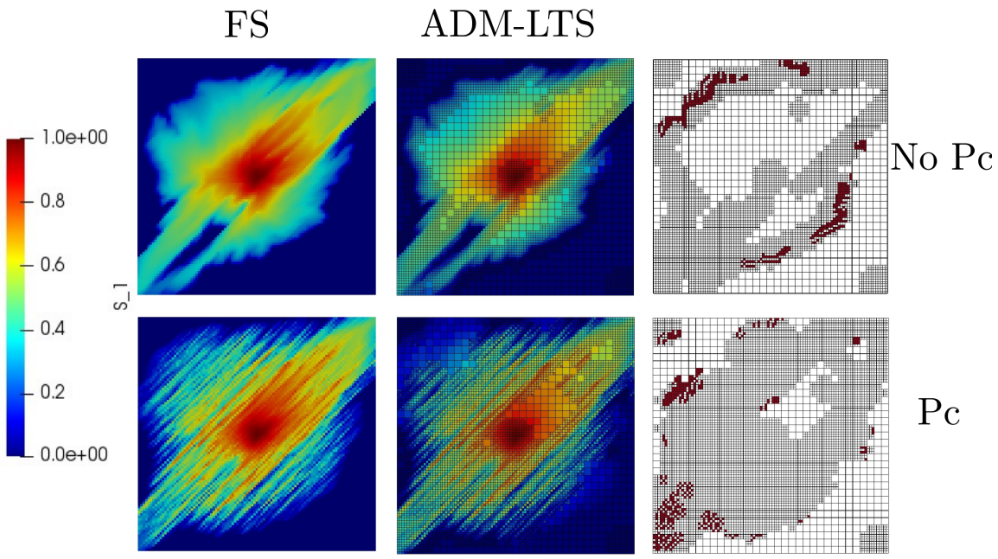


**Fig. 5.36.:** Test case 7: Reference solutions (left) and ADM-LTS saturation maps (right) without (top) and with (bottom) capillary effects.

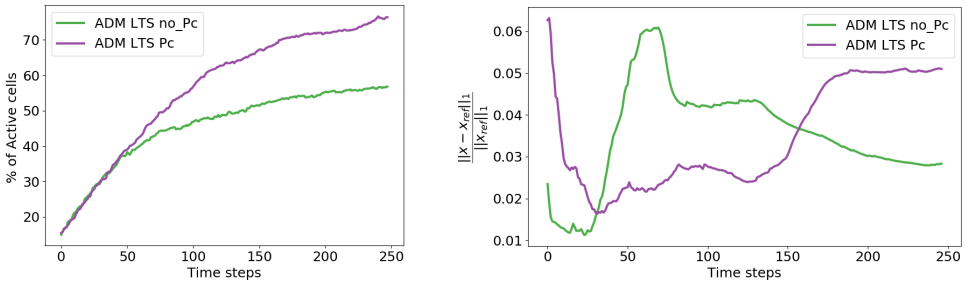


**Fig. 5.37.:** Test case 7 - Number of active cells expressed as percentage of fine grid cells (left) and saturation relative errors in $l^1$-norm (right) for the ADM-LTS simulations.

the end of the simulation, the errors start to increase. This is due to the fact that the ADM-LTS method used some coarse-scale resolution where the front appear smearing out the solution and increasing the errors.

# Conclusions and future perspectives

# 6

In this thesis we developed a novel class of conservative multirate schemes and we applied them in the context of multiphase flow in porous media. We focused, in particular, on numerical models for subsurface flow at the Darcy scale where heterogeneous permeability field and the presence of fast and slow processes, would force to use very high resolution both in space and in time if tackled with standard discretization schemes. The conservative implicit multirate method here presented is able to integrate in time with different time step sizes in different areas of the domain, detecting automatically where smaller time steps are necessary to improve accuracy. The combination with advanced schemes that employ a multiscale grid in space provide an effective methodology that reduces the CPU cost without degrading accuracy.

The conservative implicit multirate method has been presented in Chapter 2 for the time integration of hyperbolic problems. As basic implicit solver we used the TR-BDF2 method, which is a second order method, but the approach can be easily generalized to other implicit methods. The partition of fast and slow components is based on the numerical flux, in order to preserve the conservative nature of the spatial discretization employed. A consistency and stability analysis has been carried out, showing that the punctual inconsistency only arises at the interface between refined and non refined regions but, thanks to the self-adjusting strategy, the multirate scheme recovers consistency. We also proved that the multirate scheme is weakly consistent for both the fixed sub-refinement strategy and the self-adjusting strategy. We have tested this approach on several scalar equations and, to the best of our knowledge, for the first time, we have applied a self-adjusting multirate method to systems of non-linear conservation laws. The results show that our approach captures the behavior of the solution automatically and refines only where it is necessary, thus achieving a reduction of the CPU costs without significant losses of accuracy.

In Chapter 4, we illustrate a novel multirate multiscale method for coupled flow and transport equation in heterogeneous porous media developed in collaboration with Prof. Hadi Hajibeygi of TU Delft. To control the errors in time and space, an iterative multiscale strategy was used to preserve the spatial accuracy, coupled with the multirate method for temporal accuracy. The proposed approach was applied to the implicit pressure implicit saturation approximation of two-phase flow, to have the benefit of large time steps. At the same time, the flux approximation, as well as the iterative multiscale procedure, guarantees the local conservation of mass throughout the simulation. Proof-of-concept numerical tests show that pressure and saturation solutions improve compared with those obtained from the coarse-scale in time simulations, with only a small fraction of the cells being solved at the fine-scale in time. The investigations included systematic comparisons of both solution error history and computational complexity. Overall, the

proposed method allows for advancing the simulation in both space and time accounting for the intrinsic multiscale nature of the problem. As such, it develops a promising approach for large-scale multiphase simulations.

Finally, in chapter 5 a dynamic multilevel approach with a local time-stepping strategy for the solution of the transport equation in both homogeneous and heterogeneous porous media was presented. This work has been developed in collaboration with Prof. Hadi Hajibeygi and Dr. Matteo Cusini of TU Delft. The ADM-LTS method enables to capture moving fronts accurately. It combines, at each global time step, a dynamic multilevel grid in space with a local time-stepping strategy that is able to use small time step only where it is necessary – e.g., close to the moving saturation fronts. Compared with the classical ADM approach, the newly developed method allows to use more coarsear regions where the high gradients do not evolve in time. This method allows to reduce the size of the system in the nonlinear loop without loss of accuracy.

## 6.1 Future works

Generally, multiphase flow problems consider different types of physical assumptions depending on the nature of the application and on the different phases involved in the systems. A challenging development would be to extend the multirate techniques to more complex physics (i.e., compressibility, compositional effects). For example, if a gas phase is present, one should also take into account the compressibility into the model.

Another challenge is to improve the linear solver used at each step of the ADM-LTS approach. Indeed, so far the ADM-LTS algorithm solves the linear system using a direct method. For very large problems, even if an ADM grid is used to discretize the space, the resulting linear system may still be very large. Since the ADM employs hierarchy of nested grid at different resolutions, a "multigrid" preconditioner can be devised for an iterative solver [92].

Finally, the ADM-LTS approach uses structured Cartesian grids, for easy implementation. However, real field applications do not allow to use such regularity. Unstructured grids should be introduced in order to discretize complex geometries. The basic multirate scheme, however, would remain unchanged.

Moreover, we have focused our methodology on flow in porous media, however it may be beneficial also in other contexts where very different time scales are present in different parts of the domain.

# A conservative implicit multirate method for parabolic equations

A

The conservative implicit multirate approach described and analyzed in Chapter 2 is oriented to integrate in time nonlinear hyperbolic equations. However, when capillary effects are introduced into the multiphase flow equations, the transport equations for the saturation become a parabolic problem.

In this chapter we will test and validate the conservative implicit multirate method for parabolic equations applying it to 1D parabolic equation and to the saturation equation for two-phase flow problems in presence of capillary forces.

## A.1 Conservative implicit multirate method applied to parabolic equations

Nonlinear parabolic equations are of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot (k \nabla u) = 0 \qquad x \in \Omega, \quad t > 0, \tag{A.1}$$

where $k = k(u)$ is the nonlinear diffusion term.

If we consider scalar problems in a 1D domain, to discretize in space we can use the set of the cells $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, for $i \in \mathbb{Z}$ with $x_i$ the center of cell $I_i$ and size equal to $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. A conservative finite volume discretization yields a system of ordinary differential equations of the form

$$\frac{du_i}{dt}(t) = -\frac{1}{\Delta x_i}\left(F_{i+\frac{1}{2}}(t) - F_{i-\frac{1}{2}}(t)\right), \qquad i \in \mathbb{Z}, \quad t > 0$$

where $F_{i\pm\frac{1}{2}}$ is the semi-discrete numerical flux $(k\nabla u)$ at the interfaces $x_{i\pm\frac{1}{2}}$. Here, the algorithm showed in Section 2.3 can be easily applied to the new semi-discretized equation.

### A.1.1 The Barenblatt equation

Here, we apply the multirate approach, with the TR-BDF2 method as time integration scheme, to the nonlinear parabolic equation of gases in porous media

$$\begin{cases} \dfrac{\partial u}{\partial t} - \nabla \cdot (m u^{m-1} \nabla u) = 0 & x \in [-10, 10], \quad t \in (1, 6], \\ u(-10, t) = u(10, t) = 0 & t \in (1, 6]. \end{cases}$$

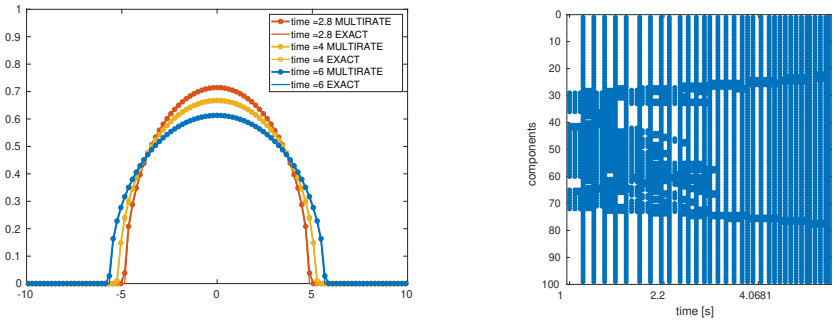


**Fig. A.1.:** Evolution of the exact (continuous line) and multirate solution (circles) Barenblatt-Pattle solution at different times (left) and set of active components at each time (right).

We focus on the so-called Barenblatt-Pattle solutions [5] that can be written in the form

$$u(x,t) = (t+t_0)^{-k}\left(A^2 - \frac{k(m-1)|x|^2}{2m(t+t_0)^{2k}}\right)_+^{\frac{1}{m-1}}$$

where $t_0$ is the initial time, $A$ is an arbitrary nonzero constant and $k = \frac{1}{m+1}$.

In this example, as numerical flux at the interfaces of each cell, we consider the following flux

$$F^{\theta}_{i+\frac{1}{2}} = \Delta t\; m\left(\frac{u^{\theta}_i + u^{\theta}_{i+1}}{2}\right)^{m-1}\left(\frac{u^{\theta}_{i+1} - u^{\theta}_i}{\Delta x}\right)$$

where $m = 3$, $A = 1$ and $t_0 = 1$. The global time step is equal to $\Delta t = 0.15$ , the number of cells are equal to $100$.

Fig. A.1 shows the multirate solution versus the exact solution at different times; the two profile are in perfect agreement. It also shows the set of active components at each simulation time. At the beginning the multirate method refines the entire area of the domain where the gas is present; at time $t = 3$[s] the multirate approach starts to refine only at the front locations.

## A.1.2 Two-phase flow in a porous media

The ADM-LTS approach presented in Chapter 5 has been applied to numerical experiments with strong effects – e.g., gravity and capillary effects. In this section, we show the conservative multirate approach applied to a two-phase flow system with capillary pressure effects. The phase viscosity ratio is equal to $\mu_{nw}/\mu_w = 10$. We consider a $500 \times 500$ [m$^2$] with $99 \times 99$ grid cells. The final time simulated is $1350$ [days] with a global time step equal to $15$ [days]. The flux tolerance for the multirate approach is $1 \times 10^{-2}$.

Fig. A.2 reports the logarithm of the heterogeneous permeability field and the wells location. The pressure constrained value are the same of the test case 5.3.7, where the values are reported in Tab. 5.1.

Fig. A.3 shows the saturation maps (left) and the active cells involved in the refinement steps of the conservative multirate approach (right) consider-

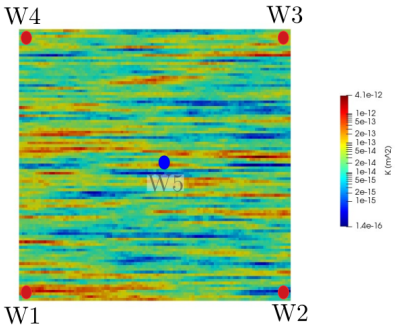


**Fig. A.2.:** Base 10 logarithm of the permeability field.

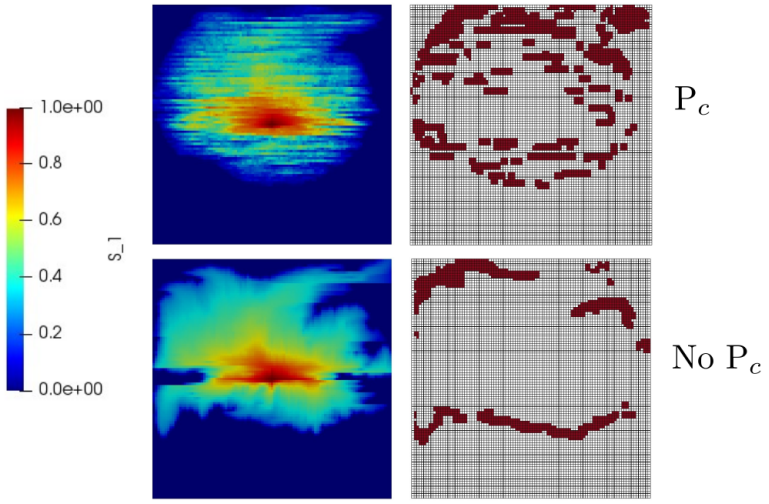


**Fig. A.3.:** Saturation maps (left) and respective active cells (right) at final time considering capillary effects (top) and without capillary pressure effects (bottom).

ing the capillary pressure effects (top) and without considering the capillary forces (bottom). The two profiles are very different as well as the set of active cells employed by the multirate approach. In fact, the multirate approach uses more active cells in the first case in order to capture the capillary heterogeneity.

# Bibliography


[1]J. Aarnes and T. Y. Hou. Multiscale Domain Decomposition Methods for Elliptic Problems with High Aspect Ratios. In: *Acta Mathematicae Applicatae Sinica* 18.1 (2002), pp. 63–76. DOI: 10.1007/s102550200004 (cit. on p. 51).

[2]I. Y. Akkutlu, Y. Efendiev, M. Vasilyeva, and Y. Wang. Multiscale model reduction for shale gas transport in poroelastic fractured media. In: *Journal of Computational Physics* 353 (2018), pp. 356 –376. DOI: 10.1016/j.jcp.2017.10.023 (cit. on p. 51).

[3]J.F. Andrus. Numerical solution of systems of ordinary differential equations separated into subsystems. In: *SIAM Journal of Numerical Analysis* 16 (1979), pp. 605–611 (cit. on pp. 4, 7).

[4]K. Aziz and A. Settari. *Petroleum Reservoir Simulation*. Applied Science Publishers, 1979 (cit. on pp. 46, 49).

[5]G. I. Barenblatt. On self-similar motions of a compressible fluid in a porous medium. In: *Akad. Nauk SSSR. Prikl. Mat. Meh* 16.6 (1952), pp. 79–6 (cit. on p. 100).

[6]P. Bastian. *Numerical computation of multiphase flow in porous media*. 1999 (cit. on pp. 43, 45, 47).

[7]M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. In: *Journal of Computational Physics* 53.3 (1984), pp. 484 –512. DOI: https://doi.org/10.1016/0021-9991(84)90073-1 (cit. on pp. 3, 72).

[8]L. Bonaventura and A. Della Rocca. Unconditionally Strong Stability Preserving Extensions of the TR-BDF2 Method. In: *Journal of Scientific Computing* 70.2 (2017), pp. 859–895 (cit. on p. 27).

[9]L. Bonaventura, F. Casella, L. Delpopolo Carciopolo, and A. Ranade. A self adjusting multirate algorithm for robust time discretization of partial differential equations. In: *Computers and Mathematics with Applications* (2019). DOI: https://doi.org/10.1016/j.camwa.2019.11.023 (cit. on pp. 7, 8, 10, 15, 27, 30, 34).

[10]S. Bosma, H. Hajibeygi, M. Tene, and H. A. Tchelepi. Multiscale finite volume method for discrete fracture modeling on unstructured grids (MS-DFM). In: *Journal of Computational Physics* 351 (2017), pp. 145 –164. DOI: 10.1016/j.jcp.2017.09.032 (cit. on p. 51).

[11]F. Bouchut. *Nonlinear stability of finite volume methods for hyperbolic conservation laws and well-balanced schemes for sources*. Springer Science & Business Media, 2004 (cit. on p. 19).

[12]R. Brooks and T. Corey. Hydraulic Properties Of Porous Media. In: *Hydrology Papers, Colorado State University* 24 (1964), p. 37 (cit. on pp. 42, 43).

[13]D. R. Brouwer and J. D. Jansen. Dynamic Optimization of Water Flooding with Smart Wells Using Optimal Control Theory. In: *SPE journal* 4 (2004), pp. 391–402. DOI: https://doi.org/10.2118/78278-MS (cit. on p. 3).

[14]G. Chavent and J. Jaffré. *Mathematical models and finite elements for reservoir simulation: single phase, multiphase and multicomponent flows through porous media*. Vol. 17. Elsevier, 1986 (cit. on p. 42).

[15]Z. Chen and T. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. In: *Mathematics of Computation* 72.242 (2003), pp. 541–576 (cit. on p. 51).

[16]Zhangxin Chen, Guanren Huan, and Yuanle Ma. *Computational methods for multiphase flows in porous media*. Vol. 2. Siam, 2006 (cit. on p. 45).

[17]M.A. Christie and M.J. Blunt. *Tenth SPE comparative solution project: A comparison of upscaling techniques*. SPE Reservoir Simulation Symposium 11-14 February, 2001 (cit. on pp. 56, 87).

[18]E. T. Chung, Y. Efendiev, W. T. Leung, and S. Ye. Generalized multiscale finite element methods for space–time heterogeneous parabolic equations. In: *Computers & Mathematics with Applications* 76.2 (2018), pp. 419 –437. DOI: 10.1016/j.camwa.2018.04.028 (cit. on p. 51).

[19]D. A. Collins, L. X. Nghiem, Y-K. Li, and J. E. Grabonstotter. An Efficient Approach to Adaptive- Implicit Compositional Simulation With an Equation of State. In: *SPE Reservoir Engineering* 7.02 (1992), pp. 259–264. DOI: 10.2118/15133-PA (cit. on pp. 4, 52).

[20]E.M. Constantinescu and A. Sandu. Multirate Timestepping Methods for Hyperbolic Conservation Laws. In: *Journal of Scientific Computing* 33.3 (2007), pp. 239–278 (cit. on pp. 4, 7).

[21]A.T. Corey. *Mechanics of immiscible fluids in porous media*. Water Resources Publication, 1994 (cit. on p. 42).

[22]M. Cusini and H. Hajibeygi. „Algebraic Dynamic Multilevel (ADM) Method For Simulations Of Multiphase Flow With An Adaptive Saturation Interpolator“. In: *ECMOR XVI-16th European Conference on the Mathematics of Oil Recovery*. 2018 (cit. on pp. 74, 85).

[23]M. Cusini, C. van Kruijsdijk, and H. Hajibeygi. Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media. In: *Journal of Computational Physics* 314 (2016), pp. 60 –79. DOI: https://doi.org/10.1016/j.jcp.2016.03.007 (cit. on pp. 3, 72, 74).

[24]M. Cusini, B. Fryer, C. van Kruijsdijk, and H. Hajibeygi. Algebraic dynamic multilevel method for compositional flow in heterogeneous porous media. In: *Journal of Computational Physics* 354 (2018), pp. 593 –612. DOI: 10.1016/j.jcp.2017.10.052 (cit. on pp. 3, 51, 95).

[25]M. Cusini, A. A. Lukyanov, J. Natvig, and H. Hajibeygi. Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. In: *Journal of Computational Physics* 299 (2015), pp. 472 –486. DOI: 10.1016/j.jcp.2015.07.019 (cit. on p. 51).

[26]M. Cusini, R. Gielisse, H. Groot, C. van Kruijsdijk, and H. Hajibeygi. Incomplete mixing in porous media: Todd-Longstaff upscaling approach versus a dynamic local grid refinement method. In: *Computational Geosciences* 23.2 (2019), pp. 373–397. DOI: 10.1007/s10596-018-9802-0 (cit. on pp. 71, 72).

[27]N.H. Darman, G. E. Pickup, and K. S. Sorbie. A comparison of two-phase dynamic upscaling methods based on fluid potentials. In: *Computational Geosciences* 6.1 (2002), pp. 5–27 (cit. on p. 3).

[28]L. Delpopolo Carciopolo, L. Bonaventura, A. Scotti, and L. Formaggia. A conservative implicit multirate method for hyperbolic problems. In: *Computational Geosciences* 23.4 (2019), pp. 647–664. DOI: 10.1007/s10596-018-9764-2 (cit. on p. 7).

[29]L. Delpopolo Carciopolo, L. Formaggia, A. Scotti, and Hajibeygi. H. Conservative multirate multiscale simulation of multiphase flow in heterogeneous porous media. In: *Journal of Computational Physics* 404 (2020), p. 109134. DOI: https://doi.org/10.1016/j.jcp.2019.109134 (cit. on p. 51).

[30]C. J. van Duijn, H. Eichel, R. Helmig, and I. S. Pop. Effective equations for two-phase flow in porous media: the effect of trapping on the microscale. In: *Transport in Porous Media* 69.3 (2007), pp. 411–428. DOI: 10.1007/s11242-006-9089-9 (cit. on p. 3).

[31]L. J. Durlofsky. Upscaling and gridding of fine scale geological models for flow simulation. In: *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy* 2024 (2005), pp. 1–59 (cit. on pp. 3, 71).

[32]Y. Efendiev and T. Y. Hou. *Multiscale finite element methods: theory and applications*. Vol. 4. Springer Science & Business Media, 2009 (cit. on p. 51).

[33]Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. In: *Journal of Computational Physics* 220.1 (2006), pp. 155 –174. DOI: 10.1016/j.jcp.2006.05.015 (cit. on pp. 3, 51).

[34]H. Eichel, R. Helmig, I. Neuweiler, and O. A. Cirpka. „Upscaling of Two-Phase Flow Processes in Porous Media“. In: *Upscaling Multiphase Flow in Porous Media: From Pore to Core and Beyond*. Dordrecht: Springer Netherlands, 2005, pp. 237–257. DOI: 10.1007/1-4020-3604-3_12 (cit. on p. 3).

[35]Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. „Finite volume methods“. In: *Handbook of numerical analysis*. Ed. by J.L. Lions and P. Ciarlet. Vol. 7. Elsevier, 2000, pp. 713–1018 (cit. on p. 19).

[36]B. Faigle, R. Helmig, I. Aavatsmark, and B. Flemisch. Efficient multiphysics modelling with adaptive grid refinement using a MPFA method. In: *Computational Geosciences* 18.5 (2014), pp. 625–636. DOI: 10.1007/s10596-014-9407-1 (cit. on pp. 3, 72).

[37]P.K. Fok. A Linearly Fourth Order Multirate Runge–Kutta Method with Error Control. In: *Journal of Scientific Computing* (2015), pp. 1–19 (cit. on pp. 4, 7, 10, 15).

[38]P.A. Forsyth and P.H. Sammon. Practical considerations for adaptive implicit methods in reservoir simulation. In: *Journal of Computational Physics* 62.2 (1986), pp. 265 –281. DOI: 10.1016/0021-9991(86)90127-0 (cit. on pp. 4, 52).

[39]L. S-K Fung, D. A. Collins, and L. X. Nghiem. An Adaptive-Implicit Switching Criterion Based on Numerical Stability Analysis (includes associated paper 18727 ). In: *SPE Reservoir Engineering* 4.01 (1989), pp. 45–51. DOI: 10.2118/16003-PA (cit. on pp. 4, 52).

[40]C.W. Gear and D.R. Wells. Multirate linear multistep methods. In: *BIT Numerical Mathematics* 24 (1984), pp. 484–502 (cit. on p. 7).

[41]A.E. Gill. *Atmosphere-Ocean Dynamics*. Academic Press, 1982 (cit. on p. 35).

[42]F.X. Giraldo, J.F. Kelly, and E.M. Constantinescu. Implicit-Explicit Formulations Of A Three-Dimensional Nonhydrostatic Unified Model Of The Atmosphere (NUMA). In: *SIAM Journal of Scientific Computing* 35.5 (2013), pp. 1162–1194 (cit. on p. 37).

[43]H. Hajibeygi and P. Jenny. Multiscale finite-volume method for parabolic problems arising from compressible multiphase flow in porous media. In: *Journal of Computational Physics* 228.14 (2009), pp. 5129 –5147. DOI: 10.1016/j.jcp.2009.04.017 (cit. on p. 51).

[44]H. Hajibeygi and H. A. Tchelepi. Compositional Multiscale Finite-Volume Formulation. In: *SPE Journal* 19.02 (2014), pp. 316–326. DOI: 10.2118/163664-PA (cit. on p. 51).

[45]H. Hajibeygi, D. Karvounis, and P. Jenny. A hierarchical fracture model for the iterative multiscale finite volume method. In: *Journal of Computational Physics* 230.24 (2011), pp. 8729–8743 (cit. on p. 51).

[46]H. Hajibeygi, S. H. Lee, I. Lunati, et al. Accurate and efficient simulation of multiphase flow in a heterogeneous reservoir with error estimate and control in the multiscale finite-volume framework. In: *SPE Journal* 17.04 (2012), pp. 1–071 (cit. on p. 56).

[47]H. Hajibeygi, G. Bonfigli, M. A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. In: *Journal of Computational Physics* 227.19 (2008), pp. 8604 –8621 (cit. on pp. 3, 51, 55, 72).

[48]F. P. Hamon, B. T. Mallison, and H. A. Tchelepi. Implicit Hybrid Upwinding for two-phase flow in heterogeneous porous media with buoyancy and capillarity. In: *Computer Methods in Applied Mechanics and Engineering* 331 (2018), pp. 701 –727. DOI: https://doi.org/10.1016/j.cma.2017.10.008 (cit. on p. 72).

[49]R. Helmig. *Multiphase Flow and Transport Processes in the Subsurface: A Contribution to the Modeling of Hydrosystems*. Environmental engineering. Springer, 1997 (cit. on pp. 42, 44).

[50]M.E. Hosea and L.F. Shampine. Analysis and implementation of TR-BDF2. In: *Applied Numerical Mathematics* 20 (1996), pp. 21–37 (cit. on p. 27).

[51]M. HosseiniMehr, M. Cusini, C. Vuik, and H. Hajibeygi. Algebraic dynamic multilevel method for embedded discrete fracture model (F-ADM). In: *Journal of Computational Physics* 373 (2018), pp. 324 –345. DOI: 10.1016/j.jcp.2018.06.075 (cit. on p. 52).

[52]T. Y. Hou and X.H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. In: *Journal of Computational Physics* 134.1 (1997), pp. 169–189 (cit. on pp. 3, 51, 72).

[53]W. Hundsdorfer, D. I. Ketcheson, and I. Savostianov. Error Analysis of Explicit Partitioned Runge–Kutta Schemes for Conservation Laws. In: *Journal of Scientific Computing* 63.3 (2015), pp. 633–653. DOI: 10.1007/s10915-014-9906-1 (cit. on p. 18).

[54]P. Jenny. Time Adaptive Conservative Finite Volume Method. In: *submitted* () (cit. on p. 72).

[55]P. Jenny and I. Lunati. Modeling complex wells with the multi-scale finite-volume method. In: *Journal of Computational Physics* 228 (2009), pp. 687–702 (cit. on pp. 49, 55, 72).

[56]P. Jenny, S.H. Lee, and H.A. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. In: *Journal of Computational Physics* 187.1 (2003), pp. 47 –67 (cit. on pp. 51, 72, 73).

[57]D. I. Ketcheson, C. B. MacDonald, and S. J. Ruuth. Spatially Partitioned Embedded Runge–Kutta Methods. In: *SIAM Journal on Numerical Analysis* 51.5 (2013), pp. 2887–2910 (cit. on p. 7).

[58]V. Kippe, Jørg E. Aarnes, and K.-A. Lie. A comparison of multiscale methods for elliptic problems in porous media flow. In: *Computational Geosciences* 12.3 (2008), pp. 377–398 (cit. on p. 51).

[59]H. Kobus. The role of large-scale experiments in groundwater and subsurface remediation research: The VEGAS concept and approach. In: *Groundwater and Subsurface Remediation* (1996), pp. 1–18 (cit. on p. 1).

[60]A. Kozlova, Z. Li, J. R. Natvig, et al. A Real-Field Multiscale Black-Oil Reservoir Simulator. In: *SPE Journal* 21.06 (2016), pp. 2049–2061. DOI: 10.2118/173226-PA (cit. on p. 51).

[61]F. Kwok and H. Tchelepi. Potential-based reduced Newton algorithm for nonlinear multiphase flow in porous media. In: *Journal of Computational Physics* 227.1 (2007), pp. 706 –727. DOI: 10.1016/j.jcp.2007.08.012 (cit. on p. 52).

[62]P. Lax and B. Wendroff. Systems of conservation laws. In: *Communications on Pure and Applied mathematics* 13.2 (1960), pp. 217–237 (cit. on p. 4).

[63]S.H. Lee, C. Wolfsteiner, and H.A. Tchelepi. Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. In: *Computational Geosciences* 12.3 (2008), pp. 351–366 (cit. on p. 51).

[64]R.J. LeVeque. *Numerical methods for conservation laws*. Vol. 132. Springer, 1992 (cit. on p. 34).

[65]M.C. Leverett. Capillary behavior in porous solids. In: *Transactions of the AIME* 142.01 (1941), pp. 152–169 (cit. on p. 43).

[66]B. Li and H.A. Tchelepi. Nonlinear analysis of multiphase transport in porous media in the presence of viscous, buoyancy, and capillary forces. In: *Journal of Computational Physics* 297 (2015), pp. 104 –131. DOI: https://doi.org/10.1016/j.jcp.2015.04.057 (cit. on p. 48).

[67]B. Li, H. A. Tchelepi, and S. M. Benson. Influence of capillary-pressure models on CO2 solubility trapping. In: *Advances in water resources* 62 (2013), pp. 488–498 (cit. on p. 44).

[68]I. Lunati, M. Tyagi, and S. H. Lee. An iterative multiscale finite volume algorithm converging to the exact solution. In: *Journal of Computational Physics* 230.5 (2011), pp. 1849 –1864 (cit. on p. 52).

[69]J. Maes, A. Moncorgé, and H. Tchelepi. Thermal adaptive implicit method: Time step selection. In: *Journal of Petroleum Science and Engineering* 106 (2013), pp. 34 –45. DOI: 10.1016/j.petrol.2013.03.019 (cit. on pp. 4, 52).

[70]B. Metz, O. Davidson, and H De Coninck. *Carbon dioxide capture and storage: special report of the intergovernmental panel on climate change*. Cambridge University Press, 2005 (cit. on p. 2).

[71]A. Moncorgé, H. A. Tchelepi, and P. Jenny. Modified sequential fully implicit scheme for compositional flow simulation. In: *Journal of Computational Physics* 337 (2017), pp. 98–115 (cit. on p. 47).

[72]A. Moncorgé, H. A. Tchelepi, and P. Jenny. Sequential fully implicit formulation for compositional simulation using natural variables. In: *Journal of Computational Physics* 371 (2018), pp. 690–711 (cit. on p. 47).

[73]J. Moortgat. Adaptive implicit finite element methods for multicomponent compressible flow in heterogeneous and fractured porous media. In: *Water Resources Research* 53 (1 2017), pp. 73–92. DOI: 10.1002/2016WR019644 (cit. on pp. 4, 52).

[74]R. J. de Moraes, J. R.P. Rodrigues, H. Hajibeygi, and J. D. Jansen. Multiscale gradient computation for flow in heterogeneous porous media. In: *Journal of Computational Physics* 336 (2017), pp. 644 –663. DOI: https://doi.org/10.1016/j.jcp.2017.02.024 (cit. on p. 3).

[75]O. Møyner and K.-A. Lie. A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. In: *Journal of Computational Physics* 304 (2016), pp. 46 –71. DOI: 10.1016/j.jcp.2015.10.010 (cit. on pp. 51, 72).

[76]O. Møyner and H. A. Tchelepi. A Mass-Conservative Sequential Implicit Multiscale Method for Isothermal Equation-of-State Compositional Problems. In: *SPE Journal* 23.06 (2018), pp. 2376–2393. DOI: 10.2118/182679-PA (cit. on p. 51).

[77]D. W. Peaceman. *Fundamentals of numerical reservoir simulation*. Vol. 6. Elsevier, 2000 (cit. on p. 45).

[78]D. W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation. In: *Society of Petroleum Engineers Journal* 18.03 (1978), pp. 183–194 (cit. on p. 49).

[79]T. Praditia, R. Helmig, and H. Hajibeygi. *Multiscale formulation for coupled flow-heat equations arising from single-phase flow in fractured geothermal reservoirs*. Computational Geosciences, 2018 (cit. on p. 51).

[80]P.J. Prince and J.R. Dormand. High order embedded Runge–Kutta formulae. In: *Journal of Computational and Applied Mathematics* 7.1 (1981), pp. 67 –75. DOI: 10.1016/0771-050X(81)90010-3 (cit. on p. 15).

[81]A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*. Vol. 37. Springer Science & Business Media, 2010 (cit. on pp. 23, 25).

[82]N. Remy, A. Boucher, and J. Wu. *Applied geostatistics with SGeMS: a user's guide*. Cambridge University Press, 2009 (cit. on p. 92).

[83]J.R. Rice. Split Runge-Kutta methods for simultaneous equations. In: *Journal of Research of the National Institute of Standards and Technology* 60 (1960) (cit. on p. 7).

[84]G. Rosatti, L. Bonaventura, A. Deponti, and G. Garegnani. An accurate and efficient semi-implicit method for section-averaged free-surface flow modelling. In: *International Journal of Numerical Methods in Fluids* 65 (2011), pp. 448–473 (cit. on p. 34).

[85]Y. Saad. *Iterative methods for sparse linear systems*. Vol. 82. siam, 2003 (cit. on p. 46).

[86]V. Savcenco, W. Hundsdorfer, and J.G. Verwer. A multirate time stepping strategy for stiff ordinary differential equations. In: *BIT Numerical Mathematics* 47 (2007), pp. 137–155 (cit. on pp. 4, 7, 8).

[87]S. Shah, O. Møyner, M. Tene, K.-A. Lie, and H. Hajibeygi. The multiscale restriction smoothed basis method for fractured porous media (F-MsRSB). In: *Journal of Computational Physics* 318 (2016), pp. 36 –57. DOI: 10.1016/j.jcp.2016.05.001 (cit. on p. 51).

[88]I. Stober and K. Bucher. *Geothermal energy: from theoretical models to exploration and development*. Springer Science & Business Media, 2013 (cit. on p. 1).

[89]M. Tene, M. S. A. Kobaisi, and H. Hajibeygi. Algebraic multiscale method for flow in heterogeneous porous media with embedded discrete fractures (F-AMS). In: *Journal of Computational Physics* 321 (2016), pp. 819 –845. DOI: 10.1016/j.jcp.2016.06.012 (cit. on p. 51).

[90]G. W. Thomas and D. H. Thurnau. Reservoir Simulation Using an Adaptive Implicit Method. In: *Society of Petroleum Engineers Journal* 23.05 (1983), pp. 759–768. DOI: 10.2118/10120-PA (cit. on pp. 4, 52).

[91]E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013 (cit. on p. 30).

[92]U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Elsevier, 2000 (cit. on p. 98).

[93]G. Tumolo and L. Bonaventura. A semi-implicit, semi-Lagrangian, DG framework for adaptive numerical weather prediction. In: *Quarterly Journal of the Royal Meteorological Society* 141 (2015), pp. 2582–2601 (cit. on p. 27).

[94]D. W. Van Batenburg, A.H. De Zwart, P. M. Boerrigter, M. Bosch, and J. C. Vink. „Application of dynamic gridding techniques to IOR/EOR processes“. In: *IOR 2011-16th European Symposium on Improved Oil Recovery*. 2011 (cit. on p. 72).

[95]M. Vasilyeva, E. T. Chung, Y. Efendiev, and J. Kim. Constrained energy minimization based upscaling for coupled flow and mechanics. In: *Journal of Computational Physics* 376 (2019), pp. 660 –674. DOI: 10.1016/j.jcp.2018.09.054 (cit. on p. 51).

[96]M. Vasilyeva, M. Babaei, E. T. Chung, and D. Spiridonov. Multiscale modeling of heat and mass transfer in fractured media for enhanced geothermal systems applications. In: *Applied Mathematical Modelling* 67 (2019), pp. 159 –178. DOI: 10.1016/j.apm.2018.10.025 (cit. on p. 51).

[97]M. Vohralík and M. F. Wheeler. A posteriori error estimates, stopping criteria, and adaptivity for two-phase flows. In: *Computational Geosciences* 17.5 (2013), pp. 789–812 (cit. on p. 56).

[98]X. Wang and H. A. Tchelepi. Trust-Region Based Solver for Nonlinear Transport in Heterogeneous Porous Media. In: *Journal of Computational Physics* 253.C (Nov. 2013), 114–137 (cit. on p. 48).

[99]Y. Wang, H. Hajibeygi, and H. A. Tchelepi. Algebraic multiscale solver for flow in heterogeneous porous media. In: *Journal of Computational Physics* 259 (2014), pp. 284 –303 (cit. on pp. 52, 54, 55).

[100]R. Younis, H. A. Tchelepi, K. Aziz, et al. Adaptively Localized Continuation-Newton Method–Nonlinear Solvers That Converge All the Time. In: *SPE Journal* 15.02 (2010), pp. 526–544 (cit. on p. 72).

[101]H. Zhou, H. A Tchelepi, et al. Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models. In: *SPE Journal* 17.02 (2012), pp. 523–539 (cit. on pp. 52, 72).

[102]P. Zitha, D. Felder R .and Zornes, K. Brown, and K. Mohanty. Increasing hydrocarbon recovery factors. In: *Society of Petroleum Engineers* (2011), pp. 1–9 (cit. on p. 3).

# Acknowledgement

I would like to thank my advisor, Prof. Luca Formaggia, for guiding me during these three years. From him I learned a lot of useful and precious knowledge, he also transferred me the passion for the research.

I am also thankful to my co-advisor, Dr. Anna Scotti, she was always present when I was in trouble helping me with very wise suggestions. She is also a very good friend, we spent a great time together in the office working at some unconventional projects.

During the period at Delft, I loved work with Prof. Hadi Hajibeygi and Dr. Matteo Cusini. Hadi always believed in my work, spurring me to do my best and improve myself. Thank you, Matteo, for the fruitful theoretical and coding discussions, and for having embraced me as a friend right away.

A really lovely thank goes to the Tender guys! Without you guys, these three years, wouldn't have been the same. I will never forget the aperitivi after work in the office, all our projects: build paper flowers, build a wooden rudder, and many others. You cheered my days when nothing was working in my research!

In the office, I met a very special person, Federica. I will bring with me all the time spent together, from the afternoons talking about research, to the Sunday lunches where we cooked delicious dishes. You have always been present to listen to my complaints and to calm me from my anxieties. She won me over so much that I had to change my passwords in "Federica ti amo", as she suggested!

I thank Tobia to be always by my side or perhaps it is better to say that I am always by his backside – as a backpack. I am deeply grateful for the help he gave me with this thesis and for listening to all my presentations. But more importantly, for having reminded me in, the hardest moments, that I could do it.

Finally, I would like to thank my parents. During my Ph.D. studies, they supported all my choices and they encouraged me when I was in my down moments. They also suffered with me for all the delays and the lost connections during conference travels.