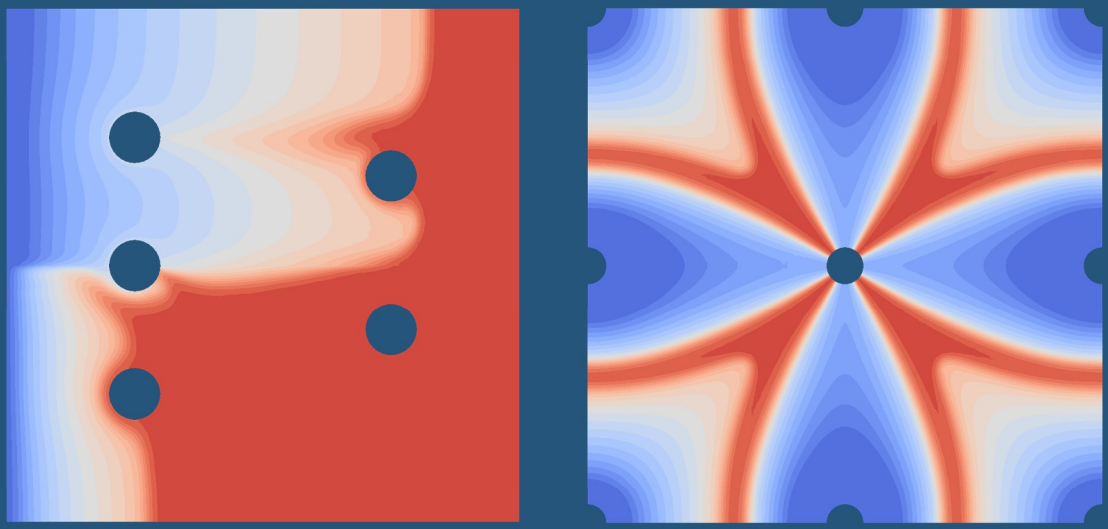
HIGH-ORDER HYBRIDIZABLE DISCONTINUOUS GALERKIN FORMULATION AND IMPLICIT RUNGE-KUTTA SCHEMES FOR MULTIPHASE FLOW THROUGH POROUS MEDIA

Albert Costa-Solé



Doctoral Thesis
Barcelona, May 2020

HIGH-ORDER HYBRIDIZABLE DISCONTINUOUS GALERKIN FORMULATION AND IMPLICIT RUNGE-KUTTA SCHEMES FOR MULTIPHASE FLOW THROUGH POROUS MEDIA

Albert Costa-Solé

Doctoral Thesis

Advisors: Josep Sarrate and Eloi Ruiz-Gironés

Barcelona, May 2020

Departament d'Enginyeria Civil i Ambiental (DECA)

Programa de Doctorat d'Enginyeria Civil

Abstract

High-order hybridizable discontinuous Galerkin formulation and implicit Runge-Kutta schemes for multiphase flow through porous media

Albert Costa-Solé

This dissertation presents high-order hybridisable discontinuous Galerkin (HDG) formulations coupled with implicit Runge-Kutta (RK) methods for the simulation of one-phase flow and two-phase flow problems.

High-order-methods can reduce the computational cost while obtaining more accurate solutions with less dissipation and dispersion errors than low order methods. HDG is an unstructured, high-order accurate, and stable method. The stability is imposed using a single parameter. In addition, it is a conservative method at the element level, which is an important feature when solving PDEs in a conservative form. Moreover, a hybridization procedure can be applied to reduce the size of the global linear system. To keep the stability and accuracy advantages in transient problems, we couple the HDG method with high-order implicit RK schemes.

The first contribution is a stable high-order HDG formulation coupled with DIRK schemes for slightly compressible one-phase flow problem. We obtain an analytical expression for the stabilization parameter using the Engquist-Osher monotone flux scheme. The selection of the stabilization parameter is crucial to ensure the stability and to obtain the high-order properties of the method. We introduce the stabilization parameter in the Newtons solver since we analytically compute its derivatives.

The second contribution is a high-order HDG formulation coupled with DIRK schemes for immiscible and incompressible two-phase flow problem. We set the water pressure and oil saturation as the main unknowns, which leads to a coupled system of two non-linear PDEs. To solve the resulting non-linear problem, we use a fix-point iterative method that alternatively solves the saturation and the pressure unknowns implicitly at each RK stage until convergence is achieved. The proposed fix-point method is memory-efficient because the saturation and the pressure are not solved at the same time.

The third contribution is a discretization scheme for the two-phase flow problem with the same spatial and temporal order of convergence. High-order spatial discretization combined with low-order temporal discretizations may lead to arbitrary

small time steps to obtain a low enough temporal error. Moreover, high-order stable DIRK schemes need a high number of stages above fourth-order. Thus, the computational cost can be severely hampered because a non-linear problem has to be solved at each RK stage. Thus, we couple the HDG formulation with high-order fully implicit RK schemes. These schemes can be unconditionally stable and achieve high-order temporal accuracy with few stages. Therefore, arbitrary large time steps can be used without hampering the temporal accuracy. We rewrite the non-linear system to reduce the memory footprint. Thus, we achieve a better sparsity pattern of the Jacobian matrix and less coupling between stages. Furthermore, we have adapted the previous fix-point iterative method. We first compute the saturation at all the stages by solving a single non-linear system using the Newton-Raphson method. Next, we solve the pressure equation sequentially at each RK stage, since it does not couple the unknowns at different stages.

The last contribution is an efficient shock-capturing method for the immiscible and incompressible two-phase flow problem to reduce the spurious oscillations that may appear in the high-order approximations of the saturation. We introduce local artificial viscosity only in the saturation equation since only the saturation variable is non-smooth. To this end, we propose a shock sensor computed from the saturation and the post-processed saturation of the HDG method. This shock sensor is computationally efficient since the post-processed saturation is computed in an element-wise manner. Our methodology allows tracking the sharp fronts as they evolve since the shock sensor is computed at all RK stages.

Acknowledgments

I started this doctoral thesis four years ago. During all this time, I have had the support of many people, whom I thank in this dissertation.

I would like to thank Josep Sarrate and Eloi Ruiz for their support. They have guided, advised and motivated me all these years. Their patience and help during all this time have made possible writing this dissertation. I would like to thank Xevi Roca for giving excellent ideas and proposing new things to do. I would like to thank all the members, the staff and the directors of LaCàN, for providing an excellent work environment.

I also thank the members of the tribunal and the reviewers of the thesis for finding the time to make defense of this dissertation possible.

I would also like to mention the funding by the "Ministerio de Economía y Competitividad" with the project CTM2014-55014-C3-3-R and the "Ministerio de Ciencia, Innovación y Universidades" with the project PGC2018-097257-B-C33. Both national projects together with the FPI-BES-2015-072833 grant have made the research possible during this time.

Thanks to all my lab-friends, Lluis, Eric and Laura who always help me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A hydrocarbon reservoir is a porous medium, which may contain water, oil and gas. Nowadays, once an exploitable hydrocarbon reservoir is found, there are three hydrocarbon recovery stages, known as primary, secondary and tertiary stages. Primary and secondary stages are responsible for around 50% of the oil production in a hydrocarbon reservoir, and the tertiary stage recovers a part of the remaining oil (Donaldson et al., 1985, 1989; Chen et al., 2006).

1. **Primary oil recovery**. In this stage, the natural pressure difference between the reservoir and the surface is high enough to move the hydrocarbon upward. This process extracts $15\% - 30\%$ of the total hydrocarbon.

2. **Secondary oil recovery** or waterflooding process consists of injecting a fluid, usually water, from one or several injection wells to mobilize the oil to the producer wells when the natural pressure difference is not high enough. Usually, after this process around 50% of the total hydrocarbon is extracted.

3. **Tertiary oil recovery** is the last stage of oil extraction. In general terms, this process aims to increase the oil miscibility and eliminate the residual oil saturation by the injection of materials that are not in the reservoir, such as chemical species.

In each stage, the nature of the fluids in the reservoir is different and, consequently, different flow models and governing equations are used (Chen et al., 2006). Multiphase flow models through porous media are essential to determine the hydrocarbon reserves, the petroleum distribution in the oilfield, and the best extraction techniques at each recovery stage (Selley and Sonnenberg, 2014). Thus, stable, accurate and robust numerical simulations are a basic tool in analyzing, planning and optimizing the hydrocarbon production. These numerical simulations have to deal with large non-linear problems, highly heterogeneous materials and complex subsurface configurations of the hydrocarbon reservoir. Several methods have been used to simulate multiphase flow through porous media, such as the the finite differences method (Aziz and Settari, 1979; Russell and Wheeler, 1983; Chen et al., 2006), the finite element method (Chen et al., 2006), the finite volume method (Chen et al., 2006; Radu et al., 2015; Salinas et al., 2017, 2018), mixed finite element methods (Chen et al., 2006; Hughes et al., 2006; Barrios et al., 2015; Hou et al., 2016; Abushaikha et al., 2017), and discontinuous Galerkin methods (Rivière and Wheeler, 2002; Rivière, 2005; Chen et al., 2006; Klieber and Rivière, 2006; Epshteyn, 2007; Ern et al., 2010; Badia and Codina, 2010; Arbogast et al., 2013; Bastian, 2014; Li and Rivière, 2015; Jamei and Ghafouri, 2016; Jamei et al., 2019).

Recently, many efforts have been focused on applying high-order methods to these kind of problems because of their advantages (Epshteyn, 2007; Li and Rivière, 2015; Fabien et al., 2018, 2020). If the analytical solution is smooth enough, then the numerical solution obtained with a method of order k converges to the analytical one as h_e^k in L^2-norm, being h_e the element size of the mesh (Babuska et al., 1981; Löhner, 2011; Wang et al., 2013). Hence, it has been shown that high-order spatial discretization methods can be more accurate than low-order ones for the same mesh resolution, that is, for the same degrees of freedom (Wang et al., 2013). Moreover, it is also reported that high-order methods introduce less diffusion and dispersion errors in the solutions (Babuska et al., 1981; Löhner, 2011; Wang et al., 2013). In addition, for the same accuracy threshold, high-order spatial discretization methods require less computational cost than low-order methods since coarser meshes can be used (Löhner, 2011; Wang et al., 2013; Löhner, 2013; Huerta et al., 2013; Giorgiani et al., 2013).

Nevertheless, to obtain these advantages in unsteady problems, the temporal integration error has to be low enough. In general, there are two ways of controlling

the temporal error. The first one is to use low-order temporal schemes with small time steps, and the second one is to use high-order temporal schemes with large time steps. However, if high temporal accuracy is required, low-order time integration schemes may require prohibitively small time steps. Thus, the computational cost of the simulation can be severely hampered because, at each time step, a non-linear problem has to be solved. In these cases, high-order temporal schemes with large time steps may alleviate the computational cost while reducing the temporal dissipation and dispersion errors (Donea and Huerta, 2003; Pazner and Persson, 2017).

To fully exploit the advantage of using arbitrary large time steps, high-order temporal schemes have to be unconditionally stable for any combination of element size, polynomial degree and time step. For instance, diagonally implicit Runge-Kutta methods (DIRK) and implicit multi-step backward differentiation formula (BDF) have been coupled with high-order spatial discretizations since they both are computationally efficient and have low memory footprint (Montlaur et al., 2012; Butcher, 2016; Fernández et al., 2018). However, DIRK schemes need to severely increase the number of stages to achieve convergence rates above fourth-order, and only BDF schemes up to second-order are unconditionally stable, known as second Dahlquist barrier (Pazner and Persson, 2017). Thus, if higher-order accuracy is required, DIRK schemes need a high number of stages, and BDF methods need small time steps. In these cases, fully implicit RK schemes may be considered (Pazner and Persson, 2017). They are unconditionally stable and achieve high-order accuracy with few stages.

Combining high-order spatial discretizations with fully implicit high-order temporal schemes increases the memory requirements. On the one hand, spatial high-order methods couple more unknowns than low-order ones for the same resolution. Thus, the Jacobian matrix involved in the non-linear system becomes denser. On the other hand, fully implicit RK schemes increases the number of unknowns and therefore, the Jacobian matrix becomes larger. Moreover, the unknowns of all the stages are coupled, which further increases the memory footprint of storing the Jacobian matrix. Therefore, specific algorithms should be devised to reduce the memory footprint for these applications.

When the solution contains sharp fronts or discontinuities, spatial high-order methods may introduce oscillations at the vicinity of the discontinuity (Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). In unsteady problems, these oscillations may not be dissipated because of the low dissipation error of high-order

temporal schemes. Thus, the accuracy of the numerical solution may be compromised as the front moves and the oscillations evolve. Moreover, the spurious oscillations may lead to a non-physical numerical solution, like negative saturation values, which hampers the robustness of the formulation. In these cases, the numerical model cannot be evaluated and the simulation has to be stopped.

1.2 Goals and layout

This dissertation aims to develop stable high-order accurate formulations in space and time for both one-phase and two-phase flow through porous media problems. Specifically, we propose a stable and high-order hybridizable discontinuous Galerkin (HDG) formulations combined with diagonally implicit and fully implicit Runge-Kutta schemes for these two problems.

High-order HDG exhibits several advantages that make it suitable for these type of simulations. First, HDG is high-order accurate. That is, it obtains a convergence rate for the scalar variables (pressure and saturation) and their corresponding gradients (flux and Darcy velocity) of order $P+1$ in L^2-norm, being P the polynomial degree, when the temporal error is low enough (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018). Moreover, an element-wise post-processing can be applied at chosen time steps to obtain a $P+2$ convergence rate for the scalar variables (Nguyen et al., 2009a,b, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018). Second, the stability is imposed through the continuity in the normal direction of a numerical flux that depends on a single stabilization parameter, τ. Third, mass is conserved at the element level. This is an important feature when solving PDEs in conservative form. Fourth, it can handle the heterogeneous reservoir properties and its geometric complexities, since unstructured meshes can be used. Fifth, the method can be hybridized in terms of the trace of the scalar variables, reducing the size of the system that has to be solved. For these reasons, high-order HDG formulation has been recently applied in porous media flow problems (Fabien et al., 2018, 2020).

To fulfill the main objective, the layout of this dissertation is organized into three main Chapters, in which we cover the following partial objectives:

1. **Developing a stable high-order HDG formulation for slightly com-**

pressible one-phase flow through porous media coupled with high-order DIRK schemes. In Chapter 3, we detail the proposed stable high-order HDG formulation coupled with DIRK schemes for slightly compressible one-phase flow through porous media. Specifically, we extend the formulation presented in (Nguyen et al., 2009b) to deal with temporal, diffusive and convective non-linear terms, and we solve the resulting non-linear system using the Newton-Raphson method. Therefore, we propose a stable convergent and high-order accurate method for both pressure and Darcy velocity. We ensure the stability of the formulation and the accuracy of the obtained solution by providing an analytical expression for the stabilization parameter using the Engquist-Osher monotone flux scheme. We highlight that the stabilitzation parameter is introduced in Newton's solver since we can analytically compute its derivatives. Note that the selection of the stabilization parameter is crucial to obtain the high-order properties of the HDG method. The stability of the method ensures the continuity of the obtained solution with respect to the problem data (initial condition, boundary condition, source term, porous media parameters,...). That is, the stability of the method ensures that small perturbations of the problem data because of small numerical errors do not lead to unbounded perturbations of the obtained solution.

2. **Developing a memory-efficient high-order HDG formulation for incompressible and immiscible two-phase flow through porous media**. In Chapters 4 and 5, we detail the proposed high-order HDG formulation coupled with high-order implicit Runge-Kutta (RK) schemes for immiscible and incompressible two-phase flow through porous media. We define the wetting pressure and non-wetting saturation as the main variables. This leads to a coupled system of two non-linear partial differential equations, the first one for the saturation and the second one for the pressure. We propose two methodologies to solve the two-phase flow problem based on different temporal discretization schemes:

   a) In Chapter 4, we present **a memory-efficient high-order HDG formulation coupled with high-order DIRK temporal schemes.** This leads to a non-linear problem at each stage of the DIRK scheme. We propose a fix-point solver, in which we alternatively solve for the saturation

unknowns, and then for the pressure unknowns implicitly until convergence is achieved at each stage of the DIRK scheme. The proposed non-linear solver reduces memory consumption since the saturation and the pressure equations are decoupled. Therefore, there is no need to solve for both unknowns at the same time.

b) In Chapter 5, we present **a memory-efficient high-order hybridizable discontinuous Galerkin (HDG) formulation coupled with high-order fully implicit Runge-Kutta schemes.** To obtain the same high-order accuracy in space and time, we propose using fully implicit high-order temporal schemes. Thus, we avoid small time steps if we use low-order schemes, or a high number of stages if we use DIRK schemes. In order to use arbitrary large time steps, we require unconditionally stable temporal schemes for any combination of element size, polynomial degree and time step. To reduce the memory footprint of coupling these spatial and temporal high-order schemes, we rewrite the non-linear system. In this way, we achieve a better sparsity pattern of the Jacobian matrix and less coupling between stages. Furthermore, we adapt the proposed fix-point iterative method in Chapter 4, which further reduces the memory consumption since it decouples the saturation and pressure equations.

3. **Developing a robust and efficient shock capturing methodology.** In Chapter 5, we introduce local artificial viscosity to reduce the spurious oscillations at the vicinity of the sharp fronts. We propose using the shock sensor introduced in Persson and Peraire (2006) to detect these oscillations. The main difference with Persson and Peraire (2006) is that we compute the shock sensor from the saturation solution and the post-processed saturation of HDG. The proposed shock sensor is computationally efficient since the post-processed saturation is computed in an element-wise manner. Thus, we do not need to solve again the problem with a different polynomial degree. Our methodology allows tracking the sharp fronts as they evolve through the different stages of the fully implicit RK scheme, since the shock sensor is computed at the stages of the RK scheme.

Chapter 2

State of the art

In this chapter, we describe the main features of the spatial and temporal discretization methods used in the numerical simulation of multiphase flow through porous media. Next, we briefly review the shock capturing methods applied in the simulation of two-phase flow problems. Finally, we summarize several non-linear solvers for the simulation of two-phase flow.

2.1 Discretization methods

During the last decades, several spatial discretization methods have been used in reservoir simulation. For instance, the finite difference method solves the differential equations by approximating the derivatives with incremental ratios. The mass is locally conserved at the discretization points (Aziz and Settari, 1979; Russell and Wheeler, 1983; Chen et al., 2006). Although this technique is fast, it requires a complex implementation to deal with the reservoir geometric complexity and to obtain high-order discretizations.

The finite volume method obtains an element-wise approximation of the solution. It can be applied to unstructured polygonal and polyhedral meshes, and therefore it has geometric flexibility. Moreover, the method is locally conservative at each element (Versteeg and Malalasekera, 2007). It has been successfully applied as a first-order method in reservoir simulations (Chen et al., 2006; Radu et al., 2015; Salinas et al., 2017, 2018).

The high-order continuous Galerkin method approximates the solution using continuous element-wise polynomials of arbitrary degree. The method can deal with the reservoir complexity since the mesh can be adapted to the geometry (Russell and Wheeler, 1983; Zienkiewicz et al., 2005; Chen et al., 2006). The number of unknowns of the linear system can be reduced by applying a hybridization procedure, and therefore the computational cost is also reduced. This method is conservative at the domain level. Also, it is an accurate method for the scalar variable that converges with a rate of $P+1$ in L^2-norm. Nevertheless, the gradient of the scalar variable loses one order of convergence in L^2-norm. Therefore, the Darcy velocity converges with order P, which is a drawback for these type of simulations.

Mixed finite element methods usually introduce the flux in the formulation as a new unknown, which is related to the gradient of the scalar variable (Chen et al., 2006; Hughes et al., 2006; Barrios et al., 2015; Hou et al., 2016; Abushaikha et al., 2017). However, it is necessary to apply different stabilization techniques for each selection of approximation spaces to obtain a stable, consistent and convergent method (Masud and Hughes, 2002; Brezzi et al., 2005; Hughes et al., 2006; Luo et al., 2011). Since they introduce a new unknown, mixed methods have more degrees of freedom than finite element methods, although a hybridization technique can also be applied to reduce the number of unknowns of the linear system.

Discontinuous Galerkin methods have been shown as competitive discretization methods in multiphase flow problems (Rivière et al., 2000; Rivière and Wheeler, 2002; Klieber and Rivière, 2006; Epshteyn and Rivière, 2006; Epshteyn, 2007; Natvig and Lie, 2008; Badia and Codina, 2010; Ern et al., 2010; Arbogast et al., 2013; Huang and Scovazzi, 2013; Bastian, 2014; Li and Rivière, 2015; Hou et al., 2016; Jamei and Ghafouri, 2016; Jamei et al., 2019). These methods are a type of mixed finite element in which the scalar and the flux are discontinuous element-by-element. Both variables also converge as $P+1$ in L^2-norm. The stability, consistency and accuracy of these methods depend on a suitable choice of a numerical flux that depends on several parameters (Bassi and Rebay, 1997). Moreover, mass conservation is verified at the element level, which is an advantage when solving PDEs in conservative form (Montlaur et al., 2008; Roca et al., 2013). These methods can use polynomials of arbitrary degree, and therefore are high-order accurate (Nguyen et al., 2013). Moreover, there are DG formulations that allow solving only for the main unknown to increase computational efficiency, which is known as primal formulation (Epshteyn,

2007).

Recently, the hybridizable discontinuous Galerkin (HDG) method has been applied in two-phase flow through porous media (Fabien et al., 2018, 2020). The HDG method has all the advantages of the discontinuous Galerkin formulations. To relate the unknowns of adjacent elements, this formulation imposes the continuity of the normal component of a numerical flux between adjacent elements. To perform this, the trace of the scalar variable is introduced on the mesh skeleton as a new variable. It approximates the scalar variables and their gradients using element-wise discontinuous polynomials (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016). Both variables also converge as $P+1$ in L^2-norm, as in DG methods. Therefore, the Darcy velocity also converges with order $P+1$. Moreover, elemental post-processing can be applied to obtain a convergence rate for the scalar variables of $P+2$ in L^2-norm (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018). Consequently, the accuracy of the obtained solutions can be increased without hampering the computational cost. The stability is imposed through the continuity in the normal direction of a numerical flux that depends on a single stabilization parameter, τ. Furthermore, mass is conserved at the element level and unstructured meshes can be used to handle the reservoir heterogeneity. Finally, this method is also hybridizable in terms of the traces, reducing the size of the global linear system.

For unsteady problems, time-marching integration schemes can be used, such as RK methods (Butcher, 1964a,b, 1996; Montlaur et al., 2012; Butcher, 2016; Pazner and Persson, 2017; Fernández et al., 2018). Given a solution at a time step, RK methods compute the solution at the next time step as a linear combination of the solution obtained at intermediate times. Those intermediate times are known as the stages of the RK method. Explicit RK schemes approximate the solution at a stage as a linear combination of the solution at previous stages. While these methods have low computational cost, they are conditionally stable, and therefore they cannot be applied with arbitrary large time steps. To use arbitrary time steps, implicit RK schemes can be used (Butcher, 1964a). For instance, in diagonally implicit Runge-Kutta (DIRK) schemes, the unknowns at the stages depend on the unknowns at the current stage and at the previous ones. Thus, the solution at the stages can be solved sequentially. However, above fourth-order, the number of stages of DIRK schemes increases faster

than the integration order (Pazner and Persson, 2017). If high-order temporal accuracy is required, the fully implicit RK schemes may be considered. These schemes are unconditionally stable and can achieve high-order temporal accuracy with few stages (Butcher, 1964b; Pazner and Persson, 2017). However, the stage unknowns are all coupled, and therefore, they involve solving a non-linear system of equations involving the unknowns of all the stages. Thus, fully implicit RK schemes have a higher memory footprint than explicit RK and DIRK methods.

2.2 Shock capturing methods

It is well known that in two-phase flow through porous media problems, the saturation can present sharp fronts (Klieber and Rivière, 2006; Arbogast et al., 2013; Jamei and Ghafouri, 2016; Jamei et al., 2019). Thus, the high-order approximation may present numerical oscillations. To reduce these spurious oscillations, shock capturing methods can be applied. The idea of these methods is to introduce additional dissipation near the sharp fronts (Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). These methods require a shock sensor to precisely detect the location of the sharp fronts. For example, in Persson and Peraire (2006), the authors propose a sensor that is computed by comparing two solutions of the problem computed with a different polynomial degree, P and $P-1$.

Slope limiting techniques are commonly applied in two-phase flow problems to reduce the spurious oscillations (Klieber and Rivière, 2006; Arbogast et al., 2013; Jamei and Ghafouri, 2016; Jamei et al., 2019). Usually these techniques reduce the order of the obtained approximations in the elements that contain the sharp fronts to linear or constant hampering the accuracy of the solution at the sharp front (Huerta et al., 2012). To overcome this drawback, h-adaptivity method is used to increase the accuracy of the simulation (Klieber and Rivière, 2006; Arbogast et al., 2013).

Shock capturing methods based on high-order non-oscillatory reconstruction schemes, known as essential non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) approaches (Shu and Osher, 1988; Qiu and Shu, 2005; Zhu et al., 2008), have been usually applied in the framework of finite differences and finite volume methods with structured meshes. Nevertheless, there is active focus on applying these schemes to unstructured meshes and improve their performance for high-order methods (Zhu et al., 2008; Farmakis et al., 2020).

Another shock capturing technique widely used in numerical simulations is the artificial viscosity method. It consists to introduce an artificial viscosity at the elements located in the vicinity of the sharp fronts (VonNeumann and Richtmyer, 1950). One of the critical aspects of this method is to compute the amount of artificial viscosity needed to reduce the spurious oscillations without changing the underlying physics of the problem. For example, in Persson and Peraire (2006), the authors relate the amount of artificial viscosity with the shock sensor and the resolution of the mesh.

2.3 Non-linear solvers for two-phase flow problems

Two-phase flow through porous media is governed by a coupled system of non-linear partial differential equations. Once the problem is discretized, a system of coupled non-linear equations involving the saturation and pressure nodal unknowns has to be solved. Robust and efficient procedures are required to solve these non-linear systems (Chen et al., 2006). The most common methodologies in two-phase flow problems that fulfil those requirements are: the simultaneous solution method (SS), also called the fully implicit method, the implicit pressure - explicit saturation (IMPES) approach, and the adaptive implicit scheme.

The SS method consists of applying the Newton-Raphson method to solve the full non-linear system of equations simultaneously. That is, solving for the saturation and the pressure at the same time. If the initial approximation is close enough, this technique achieves the convergence threshold with few non-linear iterations because of the quadratic convergence of the Newton-Raphson method. However, it has a high memory footprint, especially if we consider more complex models, such as chemical compositional flow, which has a higher number of unknowns (Chen et al., 2006). The SS method is also applied with DG formulations to solve the two-phase flow problem (Epshteyn and Rivière, 2006, 2007; Bastian, 2014).

The IMPES approach reduces the memory footprint of the simulation. It solves the non-linear system of equations by splitting it into a pressure and saturation equation. It implicitly computes the pressure by solving a linear system, while it computes the saturation with an explicit time integration scheme (Chen et al., 2004, 2006; Hurtado et al., 2006; Kou and Sun, 2010). Using an explicit temporal scheme the parameters that depend on the saturation are computed from the saturation of the previous

time step. For that reason, solving the saturation equation does not involve solving a non-linear problem at each time step. Therefore, the explicit temporal scheme reduces the computational cost per time step. However, it is conditionally stable, and therefore, arbitrary large time steps cannot be used. This stability constraint may lead to using small time steps, which are not suitable for long time simulations. To overcome this issue and taking into account that the saturation changes faster than the pressure, the authors in Chen et al. (2004, 2006); Hurtado et al. (2006) propose the improved IMPES method. This approach consists of computing the pressure with a larger time step than the saturation. This method reduces the computational cost of the simulation, but the saturation still cannot be evaluated with an arbitrarily large time step.

The IMPES approach has been implemented in DG formulations to solve the two-phase flow through porous media problem(Arbogast et al., 2013; Jamei and Ghafouri, 2016; Jamei et al., 2019). Specifically, in Jamei and Ghafouri (2016); Jamei et al. (2019), the authors present different DG formulations, and compute the saturation explicitly using a second-order total variation diminishing RK (TVD RK) scheme and a second-order Lax-Wendroff scheme, respectively.

Several works propose to compute the saturation implicitly such as Klieber and Rivière (2006); Ern et al. (2010); Fabien et al. (2018). These authors solve the saturation equation implicitly using a backward scheme. Afterwards, they update the pressure with the obtained saturation. In Klieber and Rivière (2006); Ern et al. (2010), they linearize the saturation equation by time lagging the coefficients that depend of the saturation. That is, the parameters that depend on the saturation are computed using the information of the previous time step. In Fabien et al. (2018), the authors present an HDG formulation for the spatial discretization. Furthermore, they detail a semi-implicit algorithm to solve the non-linear problem. To perform the time marching process, the authors first compute the pressure from a given saturation at a time step. Second, using the obtained pressure, they compute the saturation at the next time step by solving the non-linear saturation equation. To solve the non-linear problem they use the Newton-Raphson method. This semi-implicit process is repeated for all the time steps of the time marching process.

The adaptive implicit scheme is used to simulate the miscible displacement, black oil model, compositional flow, among others (Chen, 2000). This scheme consists to efficiently alternate the IMPES approach with the SS method in the same simulation

domain (Chen et al., 2006). That is, the SS method is only applied in the elements of a selected reservoir zone, while the IMPES approach is used in the rest of the domain. This technique reduces the computational cost of the SS approach, but it requires analyzing the stability of the time integration scheme (Chen et al., 2006).

Chapter 3

High-order HDG formulation with DIRK schemes for the simulation of one-phase flow through porous media

3.1 Introduction

In this chapter, we focus on the primary oil recovery stage, which begins when the first well is drilled and corresponds to $15\% - 30\%$ of the total oil production (Donaldson et al., 1985, 1989; Chen et al., 2006). During this stage, the pressure difference between the surface and the reservoir is high enough to move the hydrocarbon upward (Selley and Sonnenberg, 2014). One-phase flow through porous media formulation is used to model this problem (Chen et al., 2006). We consider a single Newtonian fluid that fills the voids of the porous media. Moreover, we assume slightly compressible fluid and rock. Therefore, the governing equation for this problem is a non-linear transient partial differential equation (PDE), which is obtained from the combination of the mass conservation with Darcy's law and equations of state for the fluid and the rock (Chen et al., 2006).

To obtain accurate approximations of the pressure and the Darcy velocity, several requirements have to be fulfilled. The formulation has to deal with unstructured

meshes to capture heterogeneous and complex subsurface configurations. Moreover, it has to provide highly-accurate solutions without hampering the computational cost. Therefore, unstructured high-order formulations are well-suited for these applications. Also, the formulation has to be stable and the mass should be conserved, at least, at the element level. Nowadays, several methods such as the finite differences (FD), finite volumes (FV), continuous finite elements (CG), mixed finite elements (mixed CG), and discontinuous Galerkin methods (DG), have been applied. While all of these methods have their own advantages, none of them totally fulfill the previous requirements.

Therefore, the contributions of this chapter are:

1. **To develop a high-order HDG formulation for slightly compressible one-phase flow through porous media problem coupled with high-order DIRK schemes.** We extend the formulation presented in Nguyen et al. (2009b) to deal with temporal, diffusive and convective non-linear terms. Furthermore, to obtain highly-accurate solutions in space and time, we couple the high-order spatial HDG discretization with high-order DIRK schemes. Thus, we obtain a non-linear system at each stage of the DIRK scheme that we solve sequentially, stage-by-stage, using the Newton-Raphson method.

2. **To ensure the stability and convergence of the proposed formulation with a specific choice for the stabilization parameter.** We obtain an analytical expression for the HDG stabilization parameter, which ensures the existence and uniqueness of the obtained approximation, as well as the stability and the convergence of the formulation (Kirby et al., 2012). To this end, we split the stabilization parameter into diffusive and convective parts, as in Nguyen et al. (2009a,b). We define the diffusive part according to the physical parameters of the problem. The convective part is selected using a monotone scheme flux (LeVeque, 1992; Nguyen et al., 2009a,b). Specifically, we use the Engquist-Osher monotone flux scheme, and we deduce an analytical expression of the convective part. Thus, the presented formulation provides a stable method for arbitrary polynomial degrees. We highlight that the stabilization parameter is introduced in Newton's solver since we can analytically compute its derivatives. Moreover, the proposed selection of the stabilization parameter leads to an optimally accurate high-order method. That is, the pressure and Darcy's velocity converge with an optimal convergence rate of $P + 1$.

Therefore, we propose a stable, convergent and high-order accurate method for both pressure and Darcy's velocity for slightly compressible one-phase flow through porous media problem.

The outline of the chapter is as follows. In Section 3.2, we introduce the governing equation of slightly compressible one-phase flow through porous media. In Section 3.3, we introduce the proposed HDG spatial discretization for the problem. In Section 3.4, we couple the proposed HDG formulation with DIRK schemes. In Section 3.5, we define the Netwon-Raphson method at each stage of the DIRK scheme. In Section 3.6, we detail the hybridization procedure for the Jacobian matrix and the residual. In Section 3.7, we state the local post-processing procedure. In Section 3.8, we show several examples to illustrate the advantages of the proposed formulation. Finally, in Section 3.9, we summarize the main contributions of this chapter.

3.2 Numerical model

We consider a single Newtonian fluid flow under isothermal conditions that occupies the total soil porosity. We assume that the fluid mass cannot cross the solid interface and mass fluxes can be neglected (Chen et al., 2006). Under these assumptions, the governing equations are provided by the mass conservation and the Darcy's law

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = f, \tag{3.1a}$$

$$\mathbf{v} = -\frac{1}{\mu}\mathbf{K}\left(\nabla p - \rho \mathbf{g}\right), \tag{3.1b}$$

where ϕ is the soil porosity, ρ is the hydrocarbon density, t is the time, $\mathbf{v}$ is the Darcy velocity, f is the source term, $\mathbf{K} = \text{diag}(k_{11}, k_{22}, k_{33})$ is the soil absolute permeability tensor, $\mathbf{g}$ is the gravity, and μ is the hydrocarbon viscosity.

Following Chen et al. (2006), we consider that the fluid and rock compressibility, (c_f, c_r, respectively) are constant in the pressure ranges of the simulation. Moreover, we also assume slightly compressible fluid and rock. Thus, the density and the porosity are approximated as

$$\rho \approx \rho_{ref}\left(1 + c_f(p - p_{ref})\right),$$
$$\phi \approx \phi_{ref}\left(1 + c_r(p - p_{ref})\right),$$

where ρ_{ref} and ϕ_{ref} are the reference density and the reference porosity at a reference pressure p_{ref}. Then, the governing equation becomes

$$\phi\rho c_t \frac{\partial p}{\partial t} - \nabla \cdot \left(\frac{\rho}{\mu} \mathbf{K}(\nabla p - \rho \mathbf{g}) \right) = f, \tag{3.2}$$

where $c_t = c_f + c_r$ is the total compressibility (Chen et al., 2006).

We define $\Omega \subset \mathbb{R}^d$ as a porous medium domain with boundary Γ such that $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$, where Γ_D is the Dirichlet boundary and Γ_N is the Neumann boundary. We also consider the time interval $T = (0, t_{end})$. Therefore, our numerical model for a slightly compressible one-phase flow through porous media is composed of Equation (3.2) and the corresponding boundary and initial conditions

$$\begin{cases} s(p)\dfrac{\partial p}{\partial t} + \nabla \cdot (-\mathbf{A}(p)\nabla p + \mathbf{F}(p)) = f(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ p(\mathbf{x}, t) = g_D(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_D, T), \\ (-\mathbf{A}(p)\nabla p + \mathbf{F}(p)) \cdot \mathbf{n} = g_N(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_N, T), \\ p(\mathbf{x}, 0) = p^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega, \end{cases} \tag{3.3}$$

where $g_D(\mathbf{x}, t)$ and $g_N(\mathbf{x}, t)$ are the Dirichlet and Neumann prescribed values respectively, $\mathbf{n}$ is the outward normal, $p^0(\mathbf{x})$ is the initial pressure of the reservoir, and

$$\begin{aligned} s(p) &= \phi(p)\rho(p)c_t, \\ \mathbf{A}(p) &= \frac{\rho(p)}{\mu}\mathbf{K}, \\ \mathbf{F}(p) &= \frac{\rho(p)^2}{\mu}\mathbf{K}\mathbf{g}. \end{aligned} \tag{3.4}$$

In order to introduce the HDG formulation, we rewrite Equation (3.3) as a system of first-order equations by identifying $\mathbf{q} = -\mathbf{A}(p)\nabla p$ as the diffusive flux and $\mathbf{F}(p)$ as the convective flux

$$\begin{cases} s(p)\dfrac{\partial p}{\partial t} + \nabla \cdot (\mathbf{q} + \mathbf{F}(p)) = f(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ \mathbf{q} + \mathbf{A}(p)\nabla p = 0 & \forall (\mathbf{x}, t) \in (\Omega, T), \\ p(\mathbf{x}, t) = g_D(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_D, T), \\ (\mathbf{q} + \mathbf{F}(p)) \cdot \mathbf{n} = g_N(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_N, T), \\ p(\mathbf{x}, 0) = p^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega. \end{cases} \tag{3.5}$$

3.3 Spatial discretization

We discretize the domain, Ω, with a tessellation, T_h, composed of a set of elements, e, of polynomial degree P. Afterwards, we introduce the discontinuous finite element spaces associated with the tessellation, T_h:

$$\begin{aligned}
\mathcal{V}_h^P &= \left\{ v \in L^2\left(\Omega^d\right) \;\middle|\; v_{|e} \in \left(\mathbb{S}^P\left(e\right)\right) \forall e \in \mathrm{T}_h \right\}, \\
\mathcal{W}_h^P &= \left\{ \mathbf{w} \in \left(L^2\left(\Omega^d\right)\right)^d \;\middle|\; \mathbf{w}_{|e} \in \left(\mathbb{S}^P\left(e\right)\right)^d \forall e \in \mathrm{T}_h \right\}, \\
\mathcal{M}_h^P &= \left\{ \gamma \in L^2\left(\Sigma_h\right) \;\middle|\; \gamma_{|f} \in \left(\mathbb{S}^P\left(f\right)\right) \forall f \in \Sigma_h \right\},
\end{aligned}$$

where $\mathbb{S}^P$ is the space of the polynomials of degree at most P for triangles and tetrahedra (usually denoted by $\mathbb{P}^P$), or the tensor products of polynomials of degree at most P in each coordinate direction for tensor product elements (usually denoted by $\mathbb{Q}^P$), d is the space dimension and Σ_h is the skeleton of the mesh composed of all the element faces, f. We define $\mathcal{M}_h^P\left(g_D\right) = \left\{ \gamma \in \mathcal{M}_h^P \;\middle|\; \gamma = \Pi(g_D) \text{ on } \Gamma_D \right\}$, where $\Pi(\cdot)$ is a projection operator to the space $\left\{ \gamma|_{\Gamma_D} \, \forall \gamma \in \mathcal{M}_h^P \right\}$. In this work, we use a fixed polynomial degree for all the elements. We define the scalar products on the finite element spaces:

$$\begin{aligned}
\left(u, v\right)_e &= \int_e u \; v \; \mathrm{d}\Omega \qquad && \forall u, v \in \mathcal{V}_h^P, \\
\left(\mathbf{q}, \mathbf{w}\right)_e &= \int_e \mathbf{q} \; \cdot \mathbf{w} \; \mathrm{d}\Omega \qquad && \forall \mathbf{q}, \mathbf{w} \in \mathcal{W}_h^P, \\
\left\langle \lambda, \gamma \right\rangle_{\partial e} &= \int_{\partial e} \lambda \; \gamma \; \mathrm{d}\Gamma \qquad && \forall \lambda, \gamma \in \mathcal{M}_h^P.
\end{aligned}$$

From Equation (3.5), the HDG formulation ends up with finding an approximation $\left(p_h, \mathbf{q}_h, \hat{p}_h\right) \in \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(g_D)$ such that

$$\begin{aligned}
\sum_{e \in \mathrm{T}_h} \left(\left(s\left(p_h\right) \frac{\partial p_h}{\partial t}, v\right)_e - \left(\mathbf{q}_h + \mathbf{F}\left(p_h\right), \nabla v\right)_e \right) + \\
\sum_e \left(\left\langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, \; v \right\rangle_{\partial e} - \left(f, v\right)_e \right) = 0,
\end{aligned} \tag{3.6a}$$

$$\sum_{e \in \mathrm{T}_h} \left(\left(\mathbf{A}^{-1}\left(p_h\right) \mathbf{q}_h, \mathbf{w}\right)_e - \left(p_h, \nabla \cdot \mathbf{w}\right)_e + \left\langle \hat{p}_h, \mathbf{w} \cdot \mathbf{n} \right\rangle_{\partial e} \right) = 0, \tag{3.6b}$$

$$\sum_{e \in \mathrm{T}_h} \left(\left\langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n}, \gamma \right\rangle_{\partial e} \right) - \left\langle g_N, \gamma \right\rangle_{\Gamma_N} = 0, \tag{3.6c}$$

for all $(v, \mathbf{w}, \lambda) \in \; \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(0)$, where $\hat{p}_h$ is the trace of the pressure defined on the mesh skeleton, Σ_h, and $\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h$ is the total numerical flux. Equation (3.6c) is the

transmissivity equation, in which we impose the continuity of the total numerical flux in the normal direction between adjacent elements. Therefore, this equation relates the unknowns between adjacent elements.

According to Nguyen et al. (2009a,b), we define the total numerical flux as

$$\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h = \mathbf{q}_h + \mathbf{F}\left(\hat{p}_h\right) + \tau(p_h, \hat{p}_h)(p_h - \hat{p}_h)\mathbf{n}, \qquad \text{on } \Sigma_h,$$

where τ is the stabilization parameter that depends on p_h and $\hat{p}_h$. Nevertheless, to facilitate the notation, from now on, we will not write explicitly this dependency.

Following Nguyen et al. (2009a,b), we split the τ parameter into the diffusive and convective terms as

$$\tau = \tau_{\text{diff}} + \tau_{\text{conv}},$$

and we set the diffusive and convective numerical fluxes as:

$$\hat{\mathbf{q}}_h = \mathbf{q}_h + \tau_{\text{diff}}(p_h - \hat{p}_h)\mathbf{n}, \qquad \text{on } \Sigma_h, \tag{3.7a}$$

$$\hat{\mathbf{F}}_h = \mathbf{F}\left(\hat{p}_h\right) + \tau_{\text{conv}}(p_h - \hat{p}_h)\mathbf{n}, \qquad \text{on } \Sigma_h, \tag{3.7b}$$

respectively. We define the diffusive stabilization parameter as

$$\tau_{\text{diff}} = \frac{1}{l_c}\frac{\rho(p_h)}{\mu}\gamma_{\mathbf{K}}, \tag{3.8}$$

where l_c is a characteristic length of the problem and $\gamma_{\mathbf{K}}$ is the maximum eigenvalue of the permeability tensor, $\mathbf{K}$.

To select the convective stabilization parameter, τ_{conv}, we use a monotone scheme flux, which ensures the stability of the numerical method (Nguyen et al., 2009a,b). Specifically, we define convective stabilization parameter as

$$\tau_{\text{conv}} = \frac{1}{\left(p_h - \hat{p}_h\right)^2}\int_{\hat{p}_h}^{p_h}\left(\frac{\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(s,\hat{p}_h) - \mathbf{F}(\hat{p}_h)\cdot\mathbf{n}}{p_h - \hat{p}_h}\right)\mathrm{d}s, \tag{3.9}$$

where $\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(\cdot,\cdot)$ is the Engquist-Osher monotone scheme flux (Nguyen et al., 2009a,b)

$$\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(a,b) = \frac{1}{2}\left(\mathbf{F}(a) + \mathbf{F}(b)\right)\cdot\mathbf{n} - \frac{1}{2}\int_a^b \left|\mathbf{F}'(s)\cdot\mathbf{n}\right|\mathrm{d}s. \tag{3.10}$$

It is straightforward to prove that the derivative of the convective flux is

$$\mathbf{F}'(p) = 2\frac{\rho(p)\rho_{ref}c_f}{\mu}\mathbf{K}\mathbf{g}.$$

Therefore, the sign of the integral in Equation (3.10) only depends on the product $(\mathbf{Kg})\cdot\mathbf{n}$. Thus, inserting the third equality of Equation (3.4), and Equations (3.9) and (3.10) into Equation (3.7b), we obtain the normal component of the convective numerical flux, $\hat{\mathbf{F}}_h\cdot\mathbf{n}$, as

$$\hat{\mathbf{F}}_h\cdot\mathbf{n}=\begin{cases}\dfrac{(\mathbf{Kg})\cdot\mathbf{n}}{3c_f\rho_{ref}\mu}\left(\dfrac{\rho(p_h)^3-\rho(\hat{p}_h)^3}{p_h-\hat{p}_h}\right) & \text{if } (\mathbf{Kg})\cdot\mathbf{n}\geq 0,\\ \dfrac{\rho(\hat{p}_h)^2\,(\mathbf{Kg})\cdot\mathbf{n}}{\mu} & \text{if } (\mathbf{Kg})\cdot\mathbf{n}<0.\end{cases}\tag{3.11}$$

Note that for $p_h=\hat{p}_h$, and applying the Hôpital rule to the first equation of (3.11), we verify the required property of the monotone scheme flux: $\hat{\mathbf{F}}_h\cdot\mathbf{n}(p,p)=\mathbf{F}(p)\cdot\mathbf{n}$.

Afterwards, substituting the diffusive numerical flux, Equation (3.7a), into Equation (3.6) and using Equation (3.11) as the convective numerical flux, leads to find $(p_h,\mathbf{q}_h,\hat{p}_h)\in\mathcal{V}_h^P\times\mathcal{W}_h^P\times\mathcal{M}_h^P(g_D)$ such that

$$\begin{aligned}\sum_{e\in\mathrm{T}_h}\left((s\frac{\partial p_h}{\partial t},v)_e-(\mathbf{q}_h+\mathbf{F}_h,\nabla v)_e+\langle\hat{\mathbf{F}}_h\cdot\mathbf{n},\ v\rangle_{\partial e}\right)\\ +\sum_{e\in\mathrm{T}_h}\left(\langle\mathbf{q}_h\cdot\mathbf{n}+\tau_{\text{diff}}(p_h-\hat{p}_h),\ v\rangle_{\partial e}-(f,v)_e\right)=0,\end{aligned}\tag{3.12a}$$

$$\sum_{e\in\mathrm{T}_h}\left((\mathbf{A}^{-1}\mathbf{q}_h,\mathbf{w})_e-(p_h,\nabla\cdot\mathbf{w})_e+\langle\hat{p}_h,\mathbf{w}\cdot\mathbf{n}\rangle_{\partial e}\right)=0,\tag{3.12b}$$

$$\sum_{e\in\mathrm{T}_h}\left(\langle\hat{\mathbf{F}}_h\cdot\mathbf{n},\ \gamma\rangle_{\partial e}+\langle\mathbf{q}_h\cdot\mathbf{n}+\tau_{\text{diff}}(p_h-\hat{p}_h),\ \gamma\rangle_{\partial e}\right)-\langle g_N,\gamma\rangle_{\Gamma_N}=0,\tag{3.12c}$$

for all $(v,\mathbf{w},\gamma)\in\ \mathcal{V}_h^P\times\mathcal{W}_h^P\times\mathcal{M}_h^P(0)$.

Remark 1. The HDG method is conservative at the elemental level. This is deduced from the Equation (3.12a), by setting the test function, $v=1$, in a single element and 0 in the rest

$$\underbrace{(s\frac{\partial p_h}{\partial t},1)_e}_{\text{temporal variation}}+\underbrace{\langle(\hat{\mathbf{q}}_h+\hat{\mathbf{F}}_h)\cdot\mathbf{n},\ 1\rangle_{\partial e}}_{\text{boundary inflow/outflow}}=\underbrace{(f,1)_e}_{\text{source term}},\tag{3.13}$$

that represents the mass conservation, Equation (3.1a) in integral form.

Let $\{N_i\}_{i=1,\dots,N}$ be a Lagrangian basis of shape functions of $\mathbb{S}^P(e)$, where N is the total number of element nodes, and let $\{N_l^f\}_{l=1,\dots,N_f}$ be a Lagrangian basis of shape

functions of $\mathbb{S}^P(f)$, where N_f is the total number of nodes on the element faces. We define the approximations p_h, $\mathbf{q}_h$ and $\hat{p}_h$ as

$$p_h(\mathbf{x},t) = \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N} p_i(t)N_i(\mathbf{x}), \tag{3.14}$$

$$\mathbf{q}_h(\mathbf{x},t) = \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N}\sum_{j=1}^{N_{sd}} q_{i,j}(t)N_i(\mathbf{x})\mathbf{e}_j, \tag{3.15}$$

$$\hat{p}_h(\mathbf{x},t) = \sum_{f\in \Sigma_h}\sum_{l=1}^{N_f} \hat{p}_l(t)N_l^f(\mathbf{x}), \tag{3.16}$$

where N_{sd} is the physical dimension of the problem. Similarly, the approximation of the pressure temporal derivative, $\dot{p}_h = \partial p_h/\partial t$, is defined as

$$\dot{p}_h(\mathbf{x},t) = \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N} \dot{p}_i(t)N_i(\mathbf{x}), \tag{3.17}$$

where $\dot{p}_i(t) = dp_i/dt$.

Inserting Equations (3.14), (3.15), (3.16) and (3.17) into Equation (3.12), we obtain a non-linear coupled system of first order DAE. Specifically, the problem consists of finding the coefficients $p_i(t)$, $q_{i,j}(t)$, $\dot{p}_i(t)$ for $i = 1\dots N_i$, $j = 1\dots N_{sd}$ and $\hat{p}_l(t)$ for $l = 1\dots N_f$

$$\begin{aligned}\sum_{e\in \mathrm{T}_h}\Big((s\dot{p}_h, N_i)_e - (\mathbf{q}_h + \mathbf{F}_h, \nabla N_i)_e + \langle \hat{\mathbf{F}}_h\cdot\mathbf{n}, N_i\rangle_{\partial e}\Big)\\ + \sum_e \left(\langle \mathbf{q}_h\cdot\mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h), N_i\rangle_{\partial e} - (f, N_i)_e\right) = 0,\end{aligned} \tag{3.18a}$$

$$\sum_{e\in \mathrm{T}_h}\left((\mathbf{A}^{-1}\mathbf{q}_h, N_i\mathbf{e}_j)_e - (p_h, \nabla\cdot(N_i\mathbf{e}_j))_e + \langle \hat{p}_h, N_i\mathbf{e}_j\cdot\mathbf{n}\rangle_{\partial e}\right) = 0, \tag{3.18b}$$

$$\sum_{e\in \mathrm{T}_h}\Big(\langle \hat{\mathbf{F}}_h\cdot\mathbf{n},\ N_l^f\rangle_{\partial e} + \langle \mathbf{q}_h\cdot\mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h), N_l^f\rangle_{\partial e}\Big) - \langle g_N, N_l^f\rangle_{\Gamma_N} = 0, \tag{3.18c}$$

for N_i, $N_i\mathbf{e}_j$ and N_l^f, with $i = 1\dots N_i$, $j = 1\dots N_{sd}$, $l = 1\dots N_f$.

3.4 Temporal discretization

Equation (3.18) is a DAE and we rewrite it as

$$\mathbf{R}\left(t, \mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}, \hat{\mathbf{p}}\right) = \begin{bmatrix}\mathbf{R}_{\dot{p}}\left(t, \mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}, \hat{\mathbf{p}}\right)\\ \mathbf{R}_{\mathbf{q}}\left(t, \mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}, \hat{\mathbf{p}}\right)\\ \mathbf{R}_{\hat{p}}\left(t, \mathbf{p}, \dot{\mathbf{p}}, \mathbf{q}, \hat{\mathbf{p}}\right)\end{bmatrix} = \mathbf{0}, \tag{3.19}$$

where $\mathbf{p}$, $\dot{\mathbf{p}}$, $\mathbf{q}$ and $\hat{\mathbf{p}}$ are vectors composed of all the nodal values for the pressure, $p_i(t)$, the pressure derivative, $\dot{p}_i(t)$, the numerical flux, $q_{i,j}(t)$, and the trace of the pressure, $\hat{p}_l(t)$, at time t.

Thus, given an approximation of $p_h, \dot{p}_h, \mathbf{q}_h, \hat{p}_h \in \mathcal{V}_h^P \times \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(g_D)$, $\mathbf{R}_p$, $\mathbf{R}_q$ and $\mathbf{R}_{\hat{p}}$ are defined as follows

$$
\begin{aligned}
[\mathbf{R}_{\dot{p}}]_i \equiv & \sum_{e \in \mathrm{T}_h} \left((s\dot{p}_h, N_i)_e - (\mathbf{q}_h + \mathbf{F}_h, \nabla N_i)_e + \langle \hat{\mathbf{F}}_h \cdot \mathbf{n}, N_i \rangle_{\partial e} \right) \\
& + \sum_{e \in \mathrm{T}_h} \left(\langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ N_i \rangle_{\partial e} - (f, N_i)_e \right), \\
[\mathbf{R}_q]_{i,j} \equiv & \sum_{e \in \mathrm{T}_h} \left((\mathbf{A}^{-1}\mathbf{q}_h, N_i \mathbf{e}_j)_e - (p_h, \nabla \cdot (N_i \mathbf{e}_j))_e + \langle \hat{p}_h, N_i \mathbf{e}_j \cdot \mathbf{n} \rangle_{\partial e} \right), \\
[\mathbf{R}_{\hat{p}}]_l \equiv & \sum_{e \in \mathrm{T}_h} \left(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n},\ N_l^f \rangle_{\partial e} + \langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ N_l^f \rangle_{\partial e} \right) - \langle g_N, N_l^f \rangle_{\Gamma_N}.
\end{aligned}
$$

To solve the DAE in Equation (3.19), we use a diagonally implicit Runge-Kutta (DIRK) scheme. From now on, we denote by $(\cdot)^n$ the value of any variable at time t^n, and by $(\cdot)^{n,i}$ the value of any variable at time $t^{n,i} = t^n + c^i \Delta t$, being n the time step and i the DIRK stage. Accordingly, we compute the pressure at time $t^{n+1} = t^n + \Delta t$ as

$$
\mathbf{p}^{n+1} = \mathbf{p}^n + \Delta t \sum_{i=1}^{s} c^i \dot{\mathbf{p}}^{n,i},
$$

where s is the number of stages, and $\dot{\mathbf{p}}^{n,i}$ is the approximation of $\dot{\mathbf{p}}$ at time $t^{n,i}$. The pressure at each stage of the DIRK scheme is computed as

$$
\mathbf{p}^{n,i} = \mathbf{p}^n + \Delta t \sum_{j=1}^{i} a^{ij} \dot{\mathbf{p}}^{n,j},
$$

and the $\dot{\mathbf{p}}^{n,i}$ for $i = 1, \ldots, s$ are computed as the solution of the non-linear algebraic equation

$$
\mathbf{R}\left(t^{n,i}, \mathbf{p}^n + \Delta t \sum_{j=1}^{i} a^{ij} \dot{\mathbf{p}}^{n,j}, \dot{\mathbf{p}}^{n,i}, \mathbf{q}^{n,i}, \hat{\mathbf{p}}^{n,i} \right) = \mathbf{0}. \tag{3.20}
$$

The parameters a^{ij}, b^i, c^i, with $i = 1 \ldots s$ and $j = 1 \ldots i$, define the DIRK method and are given by the Butcher's tables (Butcher, 1964a,b, 2016; Montlaur et al., 2012;

Kennedy and Carpenter, 2016; Pazner and Persson, 2017):

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b} \end{array} \equiv \begin{array}{c|cccc} c_1 & a_{11} & & & \\ c_2 & a_{21} & a_{22} & & \\ \vdots & \vdots & & \ddots & \\ c_s & a_{s1} & & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} . \tag{3.21}$$

3.5 Non-linear solver

Equation (3.20) defines a non-linear system of equations that we solve using the Newton-Raphson method at each stage of the DIRK scheme. From now on and without loss of generality, we reorder the unknowns of the non-linear system of Equation (3.20) such as

$$\mathbf{u}^{n,i} = \begin{bmatrix} \dot{\mathbf{p}}^{n,i} \\ \mathbf{q}^{n,i} \\ \hat{\mathbf{p}}^{n,i} \end{bmatrix}, \quad \text{for} \quad i = 1 \dots s.$$

Thus, the non-linear residual is

$$\mathbf{F}\left(\mathbf{u}^{n,i}\right) = \begin{bmatrix} \mathbf{R}_{\dot{p}}\left(t^{n,i}, \mathbf{p}^{n,i}, \dot{\mathbf{p}}^{n,i}, \mathbf{q}^{n,i}, \hat{\mathbf{p}}^{n,i}\right) \\ \mathbf{R}_{\mathbf{q}}\left(t^{n,i}, \mathbf{p}^{n,i}, \dot{\mathbf{p}}^{n,i}, \mathbf{q}^{n,i}, \hat{\mathbf{p}}^{n,i}\right) \\ \mathbf{R}_{\hat{p}}\left(t^{n,i}, \mathbf{p}^{n,i}, \dot{\mathbf{p}}^{n,i}, \mathbf{q}^{n,i}, \hat{\mathbf{p}}^{n,i}\right) \end{bmatrix}.$$

The Newton-Raphson method involves successive approximations of the solution $\mathbf{u}^{n,i,k}$ at i-th Runge-Kutta stage

$$\mathbf{u}^{n,i,k+1} = \mathbf{u}^{n,i,k} + \delta\mathbf{u}^{n,i,k},$$

where $\delta\mathbf{u}^{n,i,k}$ is the solution of the linear system

$$\mathbf{J}\left(\mathbf{u}^{n,i,k}\right)\delta\mathbf{u}^{n,i,k} = -\mathbf{F}\left(\mathbf{u}^{n,i,k}\right), \tag{3.22}$$

being $\mathbf{J}\left(\mathbf{u}^{n,i,k}\right)$ the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{u}^{n,i,k}$. The Jacobian matrix coefficients are detailed in Appendix A.

The process stops when an approximation is found that satisfies the prescribed tolerances

$$\begin{aligned}
\frac{\|\dot{p}_h^{n,i,k} - \dot{p}_h^{n,i,k+1}\|_{L^2(\mathrm{T}_h)}}{\|\dot{p}_h^{n,i,k+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{\dot{p}}, \qquad \|\mathbf{R}_{\dot{p}}\|_2 \leq \varepsilon_{\mathbf{F}_{\dot{p}}}, \\
\frac{\|\mathbf{q}_h^{n,i,k} - \mathbf{q}_h^{n,i,k+1}\|_{L^2(\mathrm{T}_h)}}{\|\mathbf{q}_h^{n,i,k+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{\mathbf{q}}, \qquad \|\mathbf{R}_{\mathbf{q}}\|_2 \leq \varepsilon_{\mathbf{F}_{\mathbf{q}}}, \\
\frac{\|\hat{p}_h^{n,i,k} - \hat{p}_h^{n,i,k+1}\|_{L^2(\Sigma_h)}}{\|\hat{p}_h^{n,i,k+1}\|_{L^2(\Sigma_h)}} &< \varepsilon_{\hat{p}}, \qquad \|\mathbf{R}_{\hat{p}}\|_2 \leq \varepsilon_{\mathbf{F}_{\hat{p}}},
\end{aligned} \tag{3.23}$$

where $\|\cdot\|_{L^2(\mathrm{T}_h)}$ is the norm of the $L^2(\mathrm{T}_h)$ space of functions, $\|\cdot\|_{L^2(\Sigma_h)}$ is the norm of the $L^2(\Sigma_h)$ space of functions, and $\|\cdot\|_2$ is the Euclidean norm of vectors.

3.6 Hybridization procedure

To reduce the computational cost, we hybridize the linear system in Equation (3.22) using the static condensation procedure that allows solving only for the unknowns, $\delta\hat{\mathbf{p}}^{n,i,k}$. Then, $\delta\dot{\mathbf{p}}^{n,i,k}$ and $\delta\mathbf{q}^{n,i,k}$ are obtained using an element-by-element post-process. To this end, we rewrite the linear system in Equation (3.22) as

$$\begin{bmatrix} \mathbf{J}_{\dot{p}\dot{p}}^{n,i,k} & \mathbf{J}_{\dot{p}\mathbf{q}}^{n,i,k} & \mathbf{J}_{\dot{p}\hat{p}}^{n,i,k} \\ \mathbf{J}_{\mathbf{q}\dot{p}}^{n,i,k} & \mathbf{J}_{\mathbf{q}\mathbf{q}}^{n,i,k} & \mathbf{J}_{\mathbf{q}\hat{p}}^{n,i,k} \\ \mathbf{J}_{\hat{p}\dot{p}}^{n,i,k} & \mathbf{J}_{\hat{p}\mathbf{q}}^{n,i,k} & \mathbf{J}_{\hat{p}\hat{p}}^{n,i,k} \end{bmatrix} \begin{bmatrix} \delta\dot{\mathbf{p}}^{n,i,k} \\ \delta\mathbf{q}^{n,i,k} \\ \delta\hat{\mathbf{p}}^{n,i,k} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_{\dot{p}}^{n,i,k} \\ \mathbf{R}_{\mathbf{q}}^{n,i,k} \\ \mathbf{R}_{\hat{p}}^{n,i,k} \end{bmatrix}. \tag{3.24}$$

Afterwards, we split Equation (3.24) as

$$\begin{bmatrix} \mathbf{J}_{\dot{p}\dot{p}}^{n,i,k} & \mathbf{J}_{\dot{p}\mathbf{q}}^{n,i,k} \\ \mathbf{J}_{\mathbf{q}\dot{p}}^{n,i,k} & \mathbf{J}_{\mathbf{q}\mathbf{q}}^{n,i,k} \end{bmatrix} \begin{bmatrix} \delta\dot{\mathbf{p}}^{n,i,k} \\ \delta\mathbf{q}^{n,i,k} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{\dot{p}\hat{p}}^{n,i,k} \\ \mathbf{J}_{\mathbf{q}\hat{p}}^{n,i,k} \end{bmatrix} \delta\hat{\mathbf{p}}^{n,i,k} = - \begin{bmatrix} \mathbf{R}_{\dot{p}}^{n,i,k} \\ \mathbf{R}_{\mathbf{q}}^{n,i,k} \end{bmatrix}, \tag{3.25a}$$

$$\begin{bmatrix} \mathbf{J}_{\hat{p}\dot{p}}^{n,i,k} & \mathbf{J}_{\hat{p}\mathbf{q}}^{n,i,k} \end{bmatrix} \begin{bmatrix} \delta\dot{\mathbf{p}}^{n,i,k} \\ \delta\mathbf{q}^{n,i,k} \end{bmatrix} + \mathbf{J}_{\hat{p}\hat{p}}^{n,i,k} \delta\hat{\mathbf{p}}^{n,i,k} = -\mathbf{R}_{\hat{p}}^{n,i,k}, \tag{3.25b}$$

From Equation (3.25a), we obtain

$$\begin{bmatrix} \delta\dot{\mathbf{p}}^{n,i,k} \\ \delta\mathbf{q}^{n,i,k} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\dot{p}\dot{p}}^{n,i,k} & \mathbf{J}_{\dot{p}\mathbf{q}}^{n,i,k} \\ \mathbf{J}_{\mathbf{q}\dot{p}}^{n,i,k} & \mathbf{J}_{\mathbf{q}\mathbf{q}}^{n,i,k} \end{bmatrix}^{-1} \left(- \begin{bmatrix} \mathbf{R}_{\dot{p}}^{n,i,k} \\ \mathbf{R}_{\mathbf{q}}^{n,i,k} \end{bmatrix} - \begin{bmatrix} \mathbf{J}_{\dot{p}\hat{p}}^{n,i,k} \\ \mathbf{J}_{\mathbf{q}\hat{p}}^{n,i,k} \end{bmatrix} \delta\hat{\mathbf{p}}^{n,i,k} \right). \tag{3.26}$$

Then, we substitute Equation (3.26) into Equation (3.25b), and we obtain $\delta\hat{\mathbf{p}}^{n,i,k}$ as the solution of the hybridized linear system

$$\begin{aligned}\Bigg(- \begin{bmatrix}\mathbf{J}^{n,i,k}_{\hat{p}\dot{p}} & \mathbf{J}^{n,i,k}_{\hat{p}\mathbf{q}}\end{bmatrix} \begin{bmatrix}\mathbf{J}^{n,i,k}_{\dot{p}\dot{p}} & \mathbf{J}^{n,i,k}_{\dot{p}\mathbf{q}} \\ \mathbf{J}^{n,i,k}_{\mathbf{q}\dot{p}} & \mathbf{J}^{n,i,k}_{\mathbf{q}\mathbf{q}}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{J}^{n,i,k}_{\dot{p}\hat{p}} \\ \mathbf{J}^{n,i,k}_{\mathbf{q}\hat{p}}\end{bmatrix} + \mathbf{J}^{n,i,k}_{\hat{p}\hat{p}}\Bigg)\delta\hat{\mathbf{p}}^{n,i,k} \\ = -\mathbf{R}^{n,i,k}_{\hat{p}} + \begin{bmatrix}\mathbf{J}^{n,i,k}_{\hat{p}\dot{p}} & \mathbf{J}^{n,i,k}_{\hat{p}\mathbf{q}}\end{bmatrix} \begin{bmatrix}\mathbf{J}^{n,i,k}_{\dot{p}\dot{p}} & \mathbf{J}^{n,i,k}_{\dot{p}\mathbf{q}} \\ \mathbf{J}^{n,i,k}_{\mathbf{q}\dot{p}} & \mathbf{J}^{n,i,k}_{\mathbf{q}\mathbf{q}}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{R}^{n,i,k}_{p} \\ \mathbf{R}^{n,i,k}_{\mathbf{q}}\end{bmatrix}.\end{aligned} \tag{3.27}$$

Finally, using $\delta\hat{\mathbf{p}}^{n,i,k}$, we compute $\delta\mathbf{q}^{n,i,k}$ and $\delta\dot{\mathbf{p}}^{n,i,k}$ by solving the linear system in Equation (3.26). It is important to highlight that Equation (3.26) is solved element by element, since we can reorder the terms of the matrix

$$\begin{bmatrix}\mathbf{J}^{n,i,k}_{\dot{p}\dot{p}} & \mathbf{J}^{n,i,k}_{\dot{p}\mathbf{q}} \\ \mathbf{J}^{n,i,k}_{\mathbf{q}\dot{p}} & \mathbf{J}^{n,i,k}_{\mathbf{q}\mathbf{q}}\end{bmatrix}$$

to convert it into a block diagonal matrix, where each block involves only unknowns of a single element. Thus, the computational cost of solving Equation (3.26) is low.

To solve the global system, Equation (3.27) at each stage of the Runge-Kutta method, $i = 1, \ldots, s$, and at each Newton-Raphson iteration, k, we use LU factorization.

3.7 Local post-processing

One of the main advantages of using the HDG formulation is that the pressure, p_h, and its flux, $\mathbf{q}_h$, in $\mathcal{V}_h^P$ and $\mathcal{W}_h^P$ spaces, respectively, have a rate of convergence of $P+1$ in the L^2-norm, when the temporal error is low enough. Moreover, a local post-processing can be applied to obtain a new approximation for the pressure, p_h^*, in $\mathcal{V}_h^{P+1}$ with a rate of convergence of $P+2$ in the L^2-norm (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018).

The local problem consists of finding the post-processed pressure, $p_h^* \in \mathcal{V}_h^{P+1}$ on each element, e, such that

$$(\mathbf{A}(p_h)\nabla p_h^*, \nabla v)_e = -(\mathbf{q}_h, \nabla v)_e, \tag{3.28a}$$

$$(p_h^*, 1)_e = (p_h, 1)_e, \tag{3.28b}$$

for all $v \in \mathcal{V}_h^{P+1}$.

In order to obtain a well-posed and invertible system, Equation (3.28b) is added, which imposes that the averages of the post-processed pressure, p_h^*, and the approximated pressure, p_h, are equal element by element. It is important to highlight that this procedure can be applied at selected time steps, and it is not necessary to apply it to all the time steps (Kirby et al., 2012; Nguyen et al., 2009a,b).

3.8 Examples

This section presents several examples that illustrate the capabilities of the proposed high-order HDG formulation. The first example shows a numerical evidence of the optimal convergence rates of $P+1$ in L^2-norm for the pressure, the flux and the Darcy's velocity, and also the convergence rate of $P+2$ in L^2-norm for the post-processed pressure. In addition, we check the elemental mass balance. The second example is devoted to validate our formulation and its implementation by comparing the obtained solution against an analytical one. The third example is a comparison between the solution of a fully three dimensional case against the solution of a pseudo-three dimensional case. The last example is a fully three dimensional case with three different permeability regions.

For the Examples 3.8.2 to 3.8.4, we set the characteristic length of the diffusive stabilization parameter, τ_{diff}, as 10^{-6}, see Equation (3.8). In addition, the stopping thresholds for the Newton-Raphson method are 10^{-5} for the absolute errors and 10^{-10} for the relative errors, see Equation (3.23).

For all the examples, we set $p_{ref} = p(\mathbf{x}, 0) := p^0$. We use the DIRK3-s3 scheme defined in Appendix B. All the high-order meshes are generated using the algorithms presented in Gargallo-Peiró et al. (2015, 2016); Ruiz-Gironés E.; Roca and Sarrate (2016), which are implemented in the EZ4U environment (Roca et al., 2010).

3.8.1 Convergence rate analysis

In this example, we show a numerical evidence of the convergence rates for the pressure, p_h, the flux, $\mathbf{q}_h$, the Darcy velocity $\mathbf{v}_h$, and the post-processed pressure, p_h^*. We define an analytical pressure solution

$$p = (1 + \sin(2\pi x)\sin(2\pi y))t, \tag{3.29}$$

Table 3.1: Material and fluid parameters for Example 3.8.1.

Parameter	Value	Parameter	Value
$\mathbf{K}$	$1\ \mathrm{m}^2$	μ	$1\ \mathrm{Pa}\cdot\mathrm{s}$
ϕ_{ref}	0.1	c_r	$0.01\ \mathrm{Pa}^{-1}$
ρ_{ref}	1	c_f	$0.01\ \mathrm{Pa}^{-1}$

where $x \in \Omega = (0,1) \times (0,1)$ and $t \in (0,1)$ s. We prescribe a Dirichlet boundary condition on the whole boundary and a source term in order to obtain the analytical pressure defined in Equation (3.29). We use a DIRK3-s3 scheme with $\Delta t = 0.25$ s for the time discretization. The material and fluid data used in this example are detailed in Table 3.1. We set the characteristic length of the diffusive stabilization parameter, τ_{diff}, of Equation (3.8) as 0.1. the stopping thresholds for the Newton-Raphson method are 10^{-10} for the absolute and relative errors, see Equation (3.23).

We generate a series of meshes with 16, 64, 256 and 1024 elements per side of polynomial degrees 2 to 6. Then, we measure the error in L^2-norm of the obtained approximations against the analytical solution at time 1 s.

Figures 3.1(a) and 3.1(b) show the convergence rate of the pressure and the flux in L^2-norm, respectively. We obtain the theoretically expected convergence rate of $P+1$ in L^2-norm for all the cases. Figure 3.1(c) shows the obtained L^2-error for the post-processed pressure, p_h^*, and also shows the expected convergence rates of $P+2$ for all the cases. Moreover, we also obtain a convergence rate of $P+1$ in L^2-norm for the Darcy's velocity, because it is defined in terms of the pressure and its flux, see Figure 3.1(d).

For all the meshes, we have checked the elemental mass balance, see Equation (3.13). We obtain values in the range $7.2 \cdot 10^{-10}$ to $5.3 \cdot 10^{-15}$. For instance, Figure 3.2 shows the elemental mass balance for the mesh with 256 elements and polynomial degree five.

This example validates our formulation and the selection of the numerical convective flux. That is, the proposed formulation is stable, and achieves the optimal convergence rate of $P+1$ in L^2-norm for the pressure, the flux and the Darcy velocity, and the convergence rate of $P+2$ in L^2-norm of the post-processed pressure. Furthermore, we have numerically shown the theoretical result of the elemental mass balance.

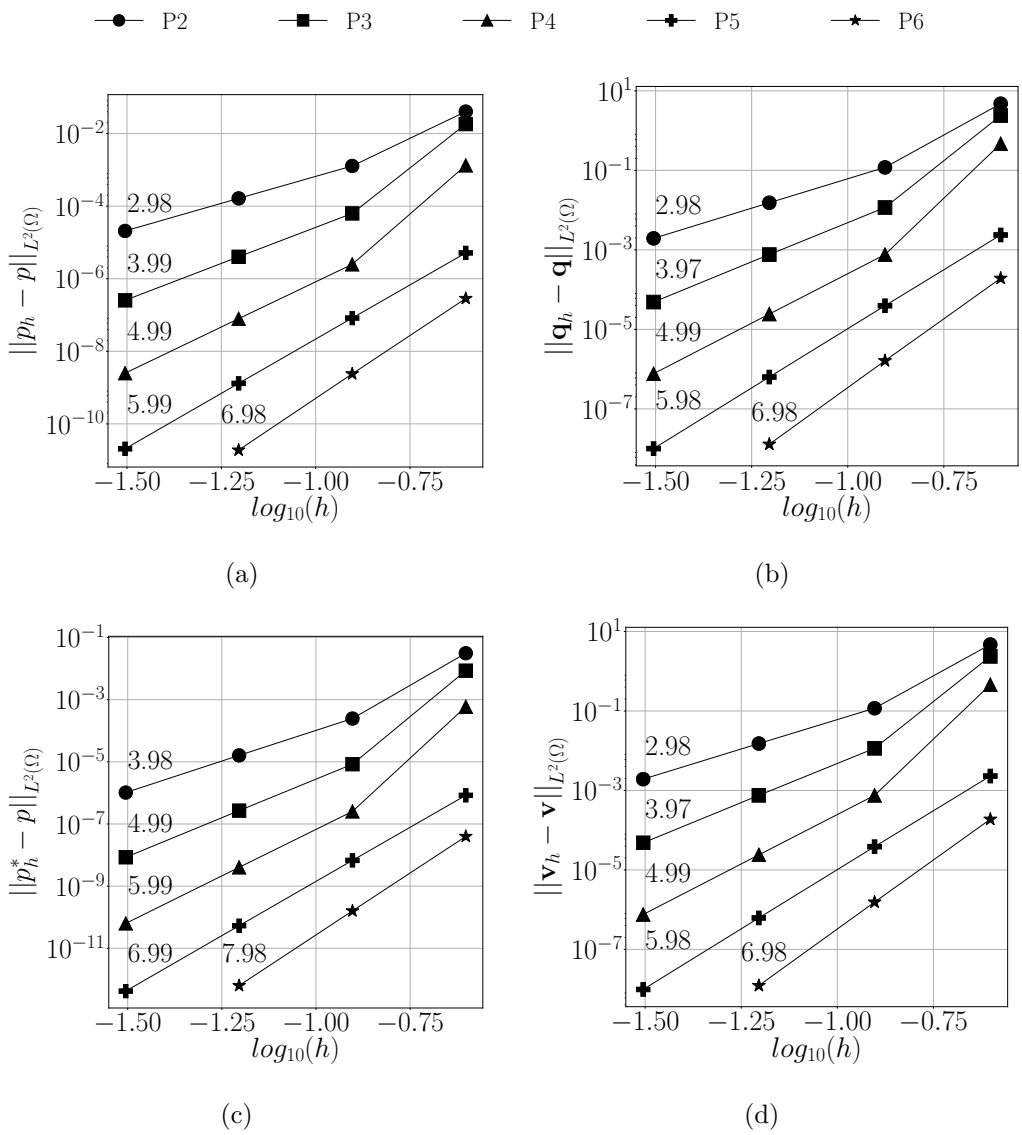


Figure 3.1: Convergence rate for: a) the pressure; b) the flux, c) the post-processed pressure and d) the Darcy velocity.

3.8.2 Comparison with an analytical solution

In this example, we compare the solution provided by our formulation with an analytical solution for a one dimensional radial flow (Chen et al., 2006). To this end, and according to Chen et al. (2006), we assume an infinite horizontal and homogeneous reservoir with constants material properties and a punctual and isolated well. Neglecting the gravity effects and using cylindrical coordinates Equation (3.2) is ex-

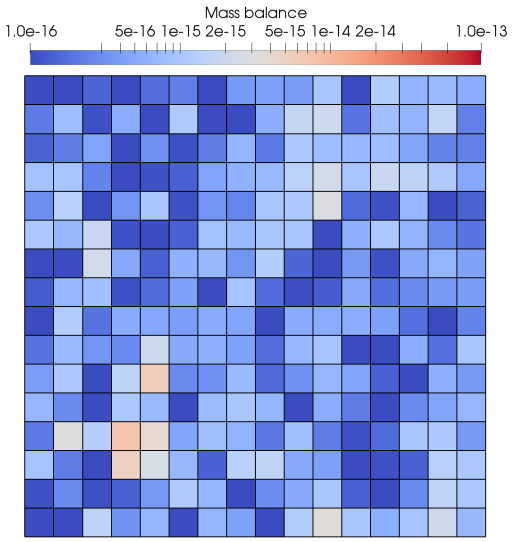


Figure 3.2: Elemental mass balance for the mesh with 256 elements and polynomial degree five.

pressed as

$$\frac{1}{\chi}\frac{\partial p}{\partial t} = \frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r}, \tag{3.30}$$

where r is the distance to the punctual well and $\chi = k/\phi\mu c_t$. Equation (3.30) is completed with the following boundary and initial conditions

$$\begin{aligned} p\left(r,t\right) &= p^0 && as\ r \to \infty, t \geq 0, \\ r\frac{\partial p}{\partial r} &= \frac{Q\mu}{2\pi kH} && as\ r \to 0, t > 0, \\ p\left(r,0\right) &= p^0 && 0 \leq r < \infty, \end{aligned}$$

where Q is the oil production well rate and H is the reservoir thickness. Under this conditions, the analytical solution of Equation (3.2) is

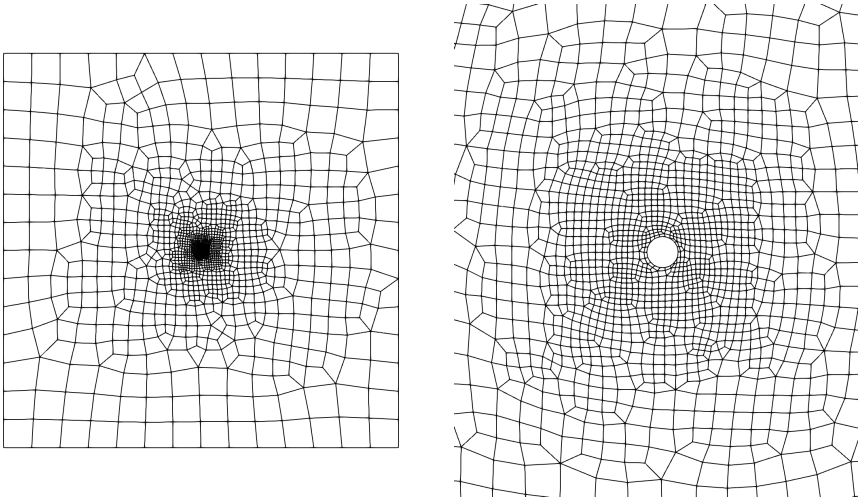
$$p\left(r,t\right) = p^0 - \frac{Q\mu}{2\pi kH} ln\left(\frac{2.25t\chi}{r^2}\right), \qquad t > 0. \tag{3.31}$$

We consider a reservoir thickness of $H = 30.48$ m and an isolated well with radius $r_w = 5.715$ cm. We use the same data as in Chen et al. (2006), which are detailed in Table 3.2.

For the numerical simulation, we define a square domain with a dimension of $\Omega = (0, 4000) \times (0, 4000)$ m, with a circular hole at the center of the domain with

Table 3.2: Material and fluid parameters for Example 3.8.2.

Parameter	Value	Parameter	Value
$\mathbf{K}$	$0.3 \cdot 10^{-13}$ m^2	p^0	244.966 atm
ϕ_{ref}	0.2	Q	0.00057742 m^3/s
c_r	$5.8 \cdot 10^{-10}$ Pa^{-1}	μ	0.00106 Pa $\cdot$ s
c_f	0 Pa^{-1}	ρ_{ref}	897.5 kg/m^3



(a) (b)

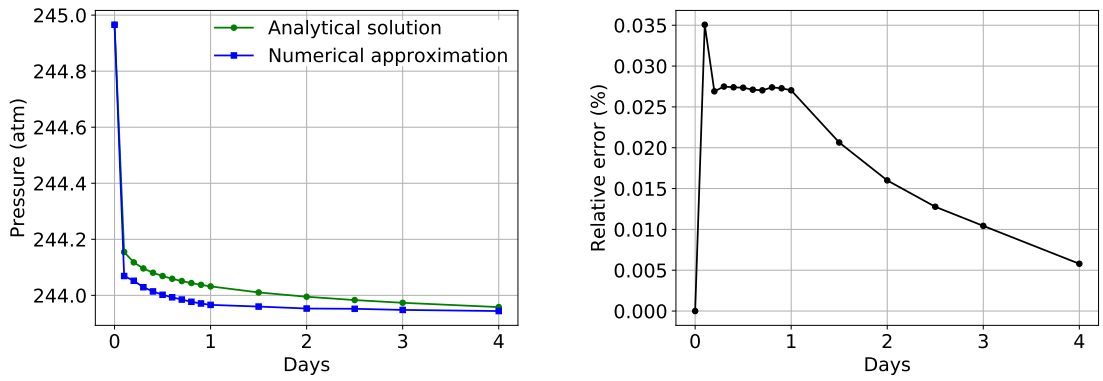
Figure 3.3: Unstructured mesh for the considered reservoir; a) global view and b) detailed view near the well.

radius $r_w = 5.715$ cm. We prescribe the pressure on the square boundary, Γ_D, and a fixed oil rate at the circular boundary, Γ_W.

The problem to be solved numerically is modeled by Equation (3.2) with a null source term, neglecting the gravitational effects, and using the following boundary and initial conditions

$$\begin{cases} \phi c_t \dfrac{\partial p}{\partial t} = \nabla \cdot \left(\dfrac{1}{\mu} \mathbf{K} \nabla p \right) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ p\left(\mathbf{x}, t\right) = 244.966 \text{ atm} & \forall (\mathbf{x}, t) \in (\Gamma_D, T), \\ \left(\dfrac{1}{\mu} \mathbf{K} \nabla p \right) \cdot \mathbf{n} = \dfrac{Q\mu}{2\pi r k H} = 0.04735 \dfrac{\text{Kg}}{\text{m}^3 s} & \forall (\mathbf{x}, t) \in (\Gamma_W, T), \\ p\left(\mathbf{x}, 0\right) = 244.966 \text{ atm} & \forall \mathbf{x} \in \Omega. \end{cases}$$

Note that the reservoir thickness, H, is introduced in our 2D model through the Neumann boundary condition.



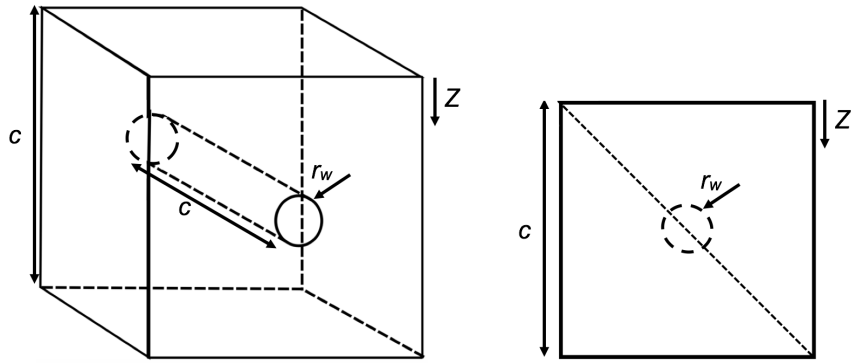
(a) (b)

Figure 3.4: a) Analytical and numerical solutions for one dimensional radial flow. b) Time evolution of the relative error.

We discretize the domain, Ω, using an unstructured mesh of 4674 quadrilateral elements of polynomial degree three, see Figures 3.3(a) and 3.3(b). The total number of nodes is 42177 and the total unknowns to be approximated with the HDG formulation is 261892. However, after applying the hybridization procedure described in Section 3.6, the size of the linear system to be solved is reduced to 37540. We simulate four days using a DIRK3-s3 scheme with $\Delta t = 0.1$ days.

Figure 3.4(a) shows the comparison between the analytical solution, Equation (3.31), and the obtained approximation at the well boundary. We observe that at the initial time step, we exactly reproduce the solution. The major difference between the analytical solution and the numerical approximation appears at time 0.1 day. This happens because the hydrocarbon is completely still and the well starts to pump at the prescribed flux rate. Afterwards, the numerical solution tends to the analytical solution.

In Figure 3.4(b), we plot the evolution of the relative error, which has the same behaviour of the approximated solution in Figure 3.4(a). At initial state, the relative error is null, because we exactly reproduce the initial and boundary conditions. Then, at time 0.1 day, there is the maximum relative error due to the overshoot of the initial pumping. Afterwards, the relative error decreases in time, until to 0.00579 % at the last time step.



(a) (b)

Figure 3.5: a) Fully three dimensional case. b) Pseudo-three dimensional case.

Table 3.3: Material and fluid parameters for Example 3.8.3. (z is the reservoir depth).

Parameter	Value	Parameter	Value
$\mathbf{K}$	$0.3 \cdot 10^{-13}$ m^2	p^0	$244.966 + 0.96z$ atm
ϕ_{ref}	0.2	f	0.58 kg/s
c_r	$5.8 \cdot 10^{-10}$ Pa^{-1}	μ	0.00106 Pa $\cdot$ s
c_f	$1.45 \cdot 10^{-9}$ Pa^{-1}	ρ_{ref}	897.5 kg/m^3

3.8.3 Fully and pseudo three dimensional comparison

This example presents a comparison between a fully three dimensional case with a pseudo-three dimensional case considering the gravity effect and a slightly compressible fluid and rock.

The physical domains, Ω^{3D} and Ω^{2D} are a cube and square (corresponding to a vertical cross section) with side $c = 50$ m, Figures 3.5(a) and 3.5(b), in which we impose no-flow condition on the outer boundaries. The parameters used for this example are defined in Table 3.3.

For the fully three dimensional case, Figure 3.5(a), we consider a cylinder well with radius $r_w = 2.0$ m and length, $l = c = 50$ m, with a central axis located at $\mathbf{x}_w = (25, y, 25)$ m. The source term, f^{3D}, is defined as

$$f^{3D} = \begin{cases} \dfrac{f}{l\pi {r_w}^2} & \text{if } \sqrt{(x_w - x)^2 + (z_w - z)^2} < r_w, \\ 0 & \text{elsewhere.} \end{cases}$$

For the pseudo-three dimensional case, Figure 3.5(b), we consider a circular well with radius $r_w = 2.0$ m and the centre located at $\mathbf{x}_w = (25, 25)$ m. The source term, f^{2D},

Table 3.4: Features for the 3D and 2D meshes of Example 3.8.3.

Item	3D mesh	2D mesh
Number of elements	512	64
Degree	3	3
Number of nodes	15625	625
Total unknowns	158720	3648
$\hat{p}$ unknowns	27548	576

is

$$f^{2D} = \begin{cases} \dfrac{f}{\pi {r_w}^2} & \text{if } \sqrt{(x_w - x)^2 + (z_w - z)^2} < r_w, \\ 0 & \text{elsewhere.} \end{cases}$$

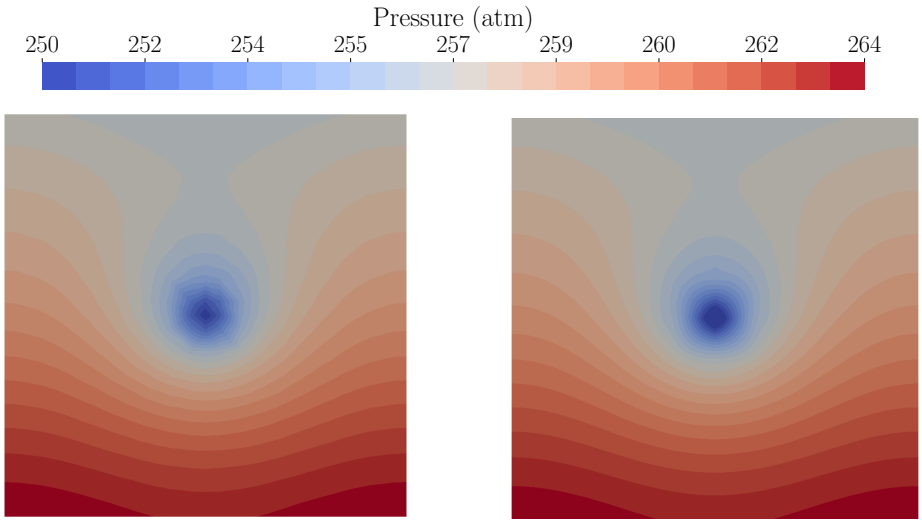
The domain, Ω, is discretized using a structured hexahedral and a quadrilateral mesh of polynomial degree three with an element size of $h = 6.25$ m, for both cases. We summarize the main features of the considered meshes in Table 3.4. For the time integration scheme, we use the DIRK3-s3 with $\Delta t = 50$ s.

Note that, we do not know a priori an initial condition compatible with the boundary condition, in which the hydrocarbon is totally still. To this end, we evolve the problem with a null source term until

$$\frac{\displaystyle\int_\Omega \|p^{n+1} - p^n\|^2 \mathrm{d}\Omega}{\displaystyle\int_\Omega 1 \mathrm{d}\Omega} < \varepsilon_{abs}, \tag{3.32}$$

where $\varepsilon_{abs} = 10^{-11}$ for that problem. To perform this, we apply the backward Euler scheme with a variable time step, $\Delta t_n = \Delta t_0 \cdot 1.105^n$, being $\Delta t_0 = 1.0$ seconds and n the step. Since we are only interested in the steady state solution, we use the backward Euler scheme because is unconditionally stable and large time steps can be used. Once the steady state is obtained, we perform the time integration using a DIRK3-s3, because we are interested in an accurate tracking of the hydrocarbon extraction process.

Figure 3.6 shows the pressure approximation at time $t = 500$ seconds. Specifically, Figure 3.6(a) corresponds to a central cross-section of the fully three dimensional case, whereas Figure 3.6(b) corresponds to the pseudo-three dimensional case. Note that we obtain similar results for both cases. This is highlighted in Figure 3.7, that shows a plot along the diagonal represented in Figure 3.5(b) of the pressure



(a) (b)

Figure 3.6: a) Pressure field: a) on the section of the fully three dimensional case, and b) of the pseudo-three dimensional case.

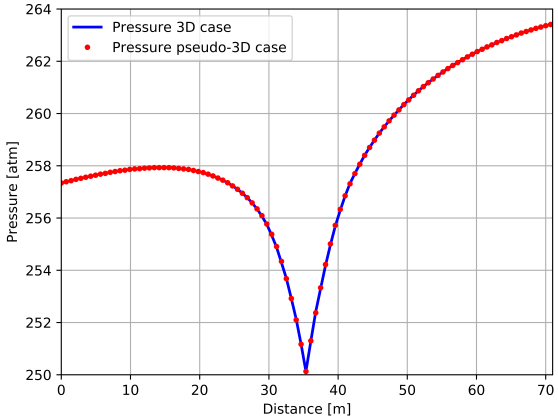


Figure 3.7: Plot over the diagonal of Figure 3.6(b) and 3.6(a).

at time $t = 500$ seconds in the fully three dimensional case and the pseudo-three dimensional case. The difference between both results is negligible and in both cases the pressure decreases near the well, because of hydrocarbon recovery, Figure 3.6. It is also important to highlight that the pressure is higher at the bottom of the reservoir than the surface due to the gravity effect.

Figure 3.8 shows the Darcy velocity vectors, Equation (3.1b), on two sections of

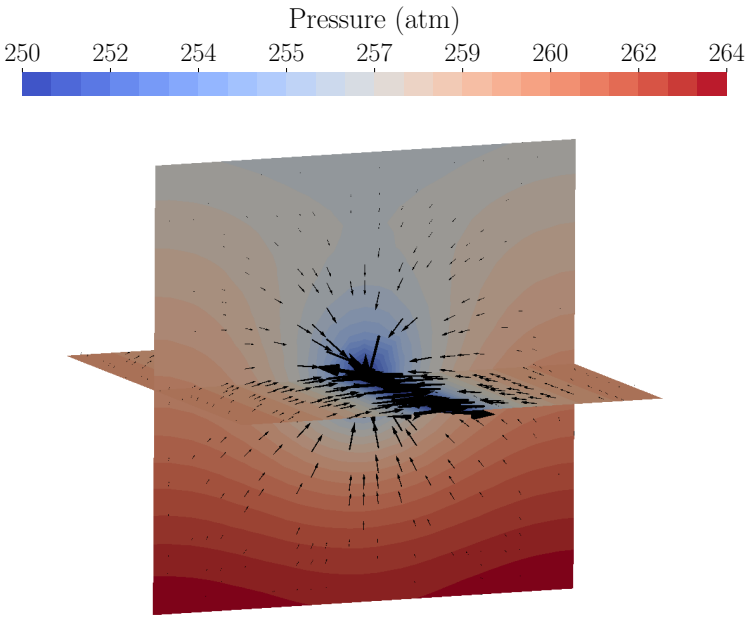


Figure 3.8: Darcy velocity vectors on two sections of the fully three dimensional case.

the fully three dimensional case. As expected, the hydrocarbon is moving towards the cylindrical well located at the center of the reservoir.

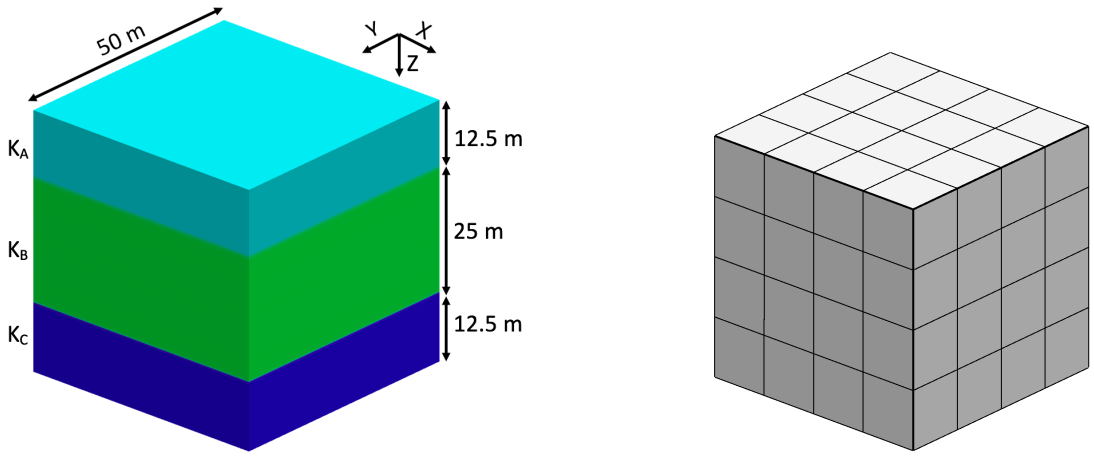
3.8.4 One-phase flow through heterogeneous material

This example corresponds to a fully three dimensional case with three different permeability regions, see Figure 3.9(a). The most permeable region is located at the middle, $\mathbf{K}_B$. At the bottom is the region with the lowest permeability, $\mathbf{K}_C$. The upper region has an intermediate permeability value, $\mathbf{K}_A$. The permeability values are detailed in Table 3.5.

The physical domain is $\Omega = (0, 50) \times (0, 50) \times (0, 50)$ m, and we impose no-flow condition on all the boundaries. We consider a spherical well with radius $r_w = 4.0$ m and the center located at $\mathbf{x}_w = (25, 25, 25)$ m. The source term is modeled as

$$f^{3D} = \begin{cases} \dfrac{f}{\frac{4}{3}\pi {r_w}^3} & \text{if } \sqrt{(x_w - x)^2 + (y_w - y)^2 + (z_w - z)^2} < r_w, \\ 0 & \text{elsewhere.} \end{cases}$$

The parameters used for this example are defined in Table 3.5.



(a) (b)

Figure 3.9: a) Permeability distribution. b) Hexahedral elements of polynomial degree four and with an element size of 12.5 m.

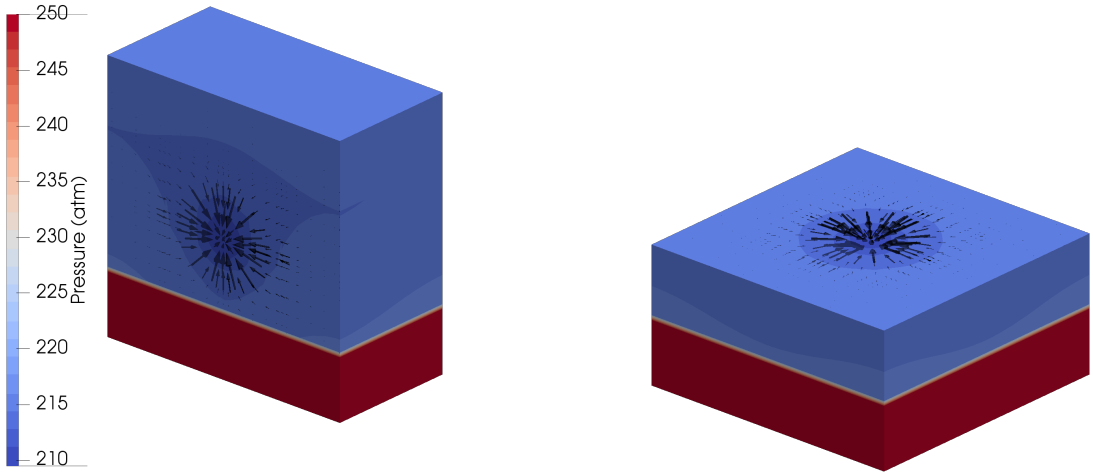
Table 3.5: Material and fluid parameters for Example 3.8.4.

Parameter	Value	Parameter	Value
$\mathbf{K}_A$	10^{-14} m^2		
$\mathbf{K}_B$	10^{-13} m^2	p^0	$244.966 + 0.96z$ atm
$\mathbf{K}_C$	10^{-17} m^2		
ϕ_{ref}	0.2	f	-2 kg/s
c_r	$5.8 \cdot 10^{-10}$ Pa^{-1}	μ	0.00106 Pa $\cdot$ s
c_f	$1.45 \cdot 10^{-9}$ Pa^{-1}	ρ_{ref}	897.5 kg/m^3

We discretize the domain, Ω, with a structured hexahedral mesh of 64 elements of polynomial degree four (4913 nodes), see Figure 3.9(b). The total number of unknowns is 38000. However, after applying the hybridization procedure, the size of the linear system to be solved is reduced to 6000. The time step for this simulation is $\Delta t = 2$ hours.

Similarly to Example 3.8.3, first we have to compute an initial condition compatible with the boundary conditions in which the hydrocarbon is totally still. For that reason, we let the system evolve until Equation (3.32) is verified. To perform this, we apply the backward Euler scheme with a variable time step and the same parameters than in Example 3.8.3.

Figure 3.10 shows the pressure field at time $t = 1$ day in two sections of the domain. The pressure increases with the depth due to the gravity effects, and it is



(a) (b)

Figure 3.10: Pressure field and Darcy velocity in two sections of the computational domain: a) YZ cross section, and b) XY cross section.

lower near the source term, because of the hydrocarbon recovery. In addition, the pressure field in the impermeable zone remains higher than in the other two regions due to the low permeability value. In the other two regions the effect of the pumping well is insignificant. Figure 3.10(b) shows a cross section perpendicular to the depth, in which the pressure field presents a circular symmetry centered at the well.

Figure 3.10 also shows that the Darcy velocity vectors are pointing to the well. Again, the different permeability regions affect to the Darcy's velocity. In the middle region, which has the highest permeability, $\mathbf{K}_B$, the fluid is moving faster than in the upper region, where the permeability is lower, $\mathbf{K}_A$, while the Darcy velocity is almost zero in the impermeable region, $\mathbf{K}_C$.

3.9 Concluding remarks

In this chapter, we have presented two contributions. In the first contribution, we have developed a high-order HDG formulation combined with high-order DIRK schemes for the one-phase flow problem through porous media. To this end, we have rewritten the initial second-order PDE as a set of first-order PDEs, and the weak form of the problem has been deduced. In the second contribution, we have deduced an analytical expression for the stabilization parameter of the proposed HDG formulation. We have

split the stabilization parameter into diffusive and convective parts. The diffusive part is selected according to the physical values of the problem, and we have used the Engquist-Osher monotone scheme flux for the convective part. These choices ensure the existence and uniqueness of the obtained approximation while providing a stable and convergent method. We have shown that the Engquist-Osher monotone scheme flux is well-suited for the one-phase flow problem. This selection allows deducing an analytical expression of the numerical convective flux. Moreover, we can introduce the numerical convective term in Newton's solver since we can analytically compute its derivatives.

We have shown in the examples the features and advantages of the proposed HDG formulation. Specifically, we have presented numerical evidence of the optimal convergence rates of $P+1$ in L^2-norm, for the pressure, the flux and the Darcy velocity, and also the convergence rate of $P+2$ in L^2-norm for the post-processed pressure. Therefore, the analytical expression for the stabilization parameter ensures the existence and uniqueness of the obtained approximation and the stability and the convergence of formulation. Moreover, we have numerically shown that the mass is conserved at the element level. We have demonstrated the capability of the proposed HDG formulation by using structured and unstructured meshes, and heterogeneous materials of the reservoir. Furthermore, we have compared the obtained approximation with an analytical solution to illustrate the accuracy of the proposed method and to validate our formulation. For the examples presented in this paper besides the convergence study, the number of iterations is around four if we consider homogeneous materials and two or three more for non-homogeneous materials.

Chapter 4

High-order HDG formulation with DIRK schemes for the simulation of two-phase flow through porous media

---

4.1 Introduction

In this chapter, we focus on the secondary oil recovery process, in which a wetting fluid is injected in the reservoir to mobilize the oil to the producer well and in this way, to maintain the production rate (Donaldson et al., 1985, 1989; Chen et al., 2006). To simulate this problem, the immiscible and incompressible two-phase flow through porous media model can be used (Chen et al., 2006). There are different formulations for this model depending on the selection of the main unknowns (Chen et al., 2006). In particular, we select the oil saturation and the water pressure as the main scalar unknowns. This leads to a coupled system of two non-linear partial differential equations, the first one for the saturation and the second one for the pressure.

As in Chapter 3, to obtain highly-accurate solutions, the proposed formulation has to fulfill several requirements. First, it has to handle with unstructured meshes to deal with the reservoir heterogeneity. Second, the formulation should be conservative

at least at the element level. This is an interesting feature when solving PDEs in a conservative form. Third, the proposed formulation has to be stable for any spatial and temporal discretization.

Note that the two-phase flow formulation has more unknowns than the one-phase flow problem, since it involves the saturation and the pressure variables. Therefore, the computational cost and the memory footprint of the simulation is higher. For that reason, it is necessary to develop and analyze specific non-linear solvers to overcome this issue. One of them is the IMPES method, which is commonly used in the industry. This approach consists of splitting the coupled non-linear system into pressure and saturation equations, and solving them with implicit and explicit schemes, respectively. The explicit schemes have low computational cost. However, these schemes are not unconditionally stable, and therefore, arbitrary large time steps cannot be used. This stability restriction may hamper the computational cost since small time steps may be required, which are not suitable for long time simulations.

Thus, the contribution of this chapter is:

1. **To develop a memory-efficient high-order HDG formulation for incompressible and immiscible two-phase flow through porous media coupled with high-order DIRK temporal schemes.** To obtain highly-accurate solutions in space and time, we couple high-order HDG formulation with DIRK schemes. This leads to an algebraic non-linear problem, which we solve at each stage using a fix-point iterative method that alternatively solves for the pressure and saturation until we achieve convergence. The proposed fix-point iterative method is memory-efficient in the sense that the saturation and pressure are not solved at the same time. As we are using DIRK schemes, we solve for the oil saturation and water pressure unknowns sequentially, stage-by-stage, which further reduces the memory consumption.

The outline of the chapter is as follows. In Section 4.2, we introduce the numerical model for the immiscible and incompressible two-phase flow through porous media problem. In Section 4.3, we deduce the corresponding HDG formulation. In Section 4.4, we couple the high-order HDG formulation with DIRK schemes. In Section 4.5, we detail the proposed non-linear solver. In Section 4.6, we state the local post-processing procedure. In Section 4.7, we show several examples to illustrate the advantages of the proposed formulation. Finally, in Section 4.8, we summarize the main contributions of this chapter.

4.2 Numerical model

The two-phase flow through porous media is characterised by a wetting phase, w, and the non-wetting phase, o. In this work, we consider water for the wetting phase and oil for the non-wetting phase. Both phases fill the voids of the soil (Chen et al., 2006; Bear and Verruijt, 2012), and verify:

$$S_w + S_o = 1,$$

where S_w and S_o are the saturation for the water and oil, respectively. Due to the interface tension between phases, there is a discontinuity in the pressure field, which is the capillary pressure, p_c:

$$p_c = p_o - p_w,$$

where p_o is the oil pressure and p_w is the water pressure.

The governing equations for two-phase flow through porous media are provided by the mass conservation for each phase:

$$\frac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \nabla\cdot(\rho_\alpha \mathbf{v}_\alpha) = \rho_\alpha f_\alpha \qquad \alpha = w, o,$$

where ϕ is the porosity of the porous media, and ρ_α, $\mathbf{v}_\alpha$ and f_α are the density the velocity and the source term of phase α, respectively. The velocity of each phase, $\mathbf{v}_\alpha$, is given by the Darcy's law as:

$$\mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K}\nabla p_\alpha \qquad \alpha = w, o,$$

where $\mathbf{K} = \mathrm{diag}(k_{11}, k_{22}, k_{33})$ is the absolute permeability and $\lambda_\alpha = k_{r\alpha}/\mu_\alpha$ is the phase mobility, being $k_{r\alpha}$ and μ_α the relative permeability and the viscosity of phase α, respectively.

In this work, the capillary pressure, p_c, and the relative permeability of each phase, $k_{r\alpha}$, are computed from the Brooks-Corey model (Corey, 1964):

$$\begin{aligned} p_c &= p_e(1 - S_{eo})^{-1/\theta}, \\ k_{rw} &= (1 - S_{eo})^{(2+3\theta)/\theta}, \\ k_{ro} &= {S_{eo}}^2 \left(1 - (1 - S_{eo})^{(2+\theta)/\theta}\right), \end{aligned} \tag{4.1}$$

where p_e is the entry pressure, θ is the pore size distribution, and

$$S_{eo} = \frac{S_o - S_{ro}}{1 - S_{rw} - S_{ro}}$$

is the effective oil saturation, being S_{ro} and S_{rw} the residual oil and water saturation, respectively.

There are several formulations to solve the two-phase flow problem. In this work, we use the water pressure, p_w, and the oil saturation, S_o, as main unknowns. According to Chen et al. (2006); Bear and Verruijt (2012), the total phase mobility and the total velocity are defined as:

$$\lambda_t = \lambda_o + \lambda_w, \qquad \mathbf{v}_t = \mathbf{v}_o + \mathbf{v}_w.$$

Assuming immiscible and incompressible fluids and incompressible rock, we obtain a system of governing equations composed of the oil saturation equation:

$$\phi \frac{\partial S_o}{\partial t} - \nabla \cdot (\lambda_o \mathbf{K}(\nabla p_c + \nabla p_w)) = f_o, \tag{4.2}$$

and the water pressure equation:

$$-\nabla \cdot (\mathbf{K}\,(\lambda_o \nabla p_c + \lambda_t \nabla p_w)) = f_o + f_w, \tag{4.3}$$

see Klieber and Rivière (2006) for additional details.

Let $\Omega \subset \mathbb{R}^d$ be a porous medium domain and $T = (0, t_{end})$ a time interval. We consider that the boundary of Ω is divided in three disjoint parts such that $\Gamma = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_{nf}$, where Γ_{in} is the inflow boundary (water is injected), Γ_{out} is the outflow boundary (water and oil are extracted) and Γ_{nf} is the no-flow boundary. Considering appropriate boundary conditions and assuming immiscible and incompressible fluids and incompressible rock, a system of two coupled non-linear partial differential equations (PDEs) is obtained (Klieber and Rivière, 2006). For the oil saturation Equation (4.2):

$$\begin{cases} \phi \dfrac{\partial S_o}{\partial t} - \nabla \cdot (\lambda_o \mathbf{K}(\nabla p_c + \nabla p_w)) = f_o(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ S_o^{\Gamma_{in}}(\mathbf{x}, t) = g_{Ds}^{in}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{in}, T), \\ \left(\dfrac{\lambda_o \lambda_w}{\lambda_t} \mathbf{K} \nabla p_c \right) \cdot \mathbf{n} = g_{Ns}^{out}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{out}, T), \\ \mathbf{v}_o \cdot \mathbf{n} = 0 & \forall (\mathbf{x}, t) \in (\Gamma_{nf}, T), \\ S_o(\cdot, 0) = S_o^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega, \end{cases} \tag{4.4}$$

where $g_{Ds}^{in}(\mathbf{x}, t)$ is the prescribed value of the saturation on the inflow boundary, $g_{Ns}^{out}(\mathbf{x}, t)$ is the prescribed value on the output boundary, and $S_o^0(\mathbf{x})$ is the initial

saturation value. For the water pressure Equation (4.3):

$$\begin{cases} -\nabla \cdot (\lambda_t \mathbf{K} \nabla p_w + \lambda_o \mathbf{K} \nabla p_c) = f_o(\mathbf{x}, t) + f_w(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ p_w^{\Gamma_{in}}(\mathbf{x}, t) = g_{Dp}^{in}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{in}, T), \\ p_w^{\Gamma_{out}}(\mathbf{x}, t) = g_{Dp}^{out}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{out}, T), \\ \mathbf{v}_t \cdot \mathbf{n} = 0 & \forall (\mathbf{x}, t) \in (\Gamma_{nf}, T), \end{cases} \tag{4.5}$$

where $g_{Dp}^{in}(\mathbf{x}, t), g_{Dp}^{out}(\mathbf{x}, t)$ are the prescribed values of the pressure on the inflow and outflow boundaries, respectively.

We rewrite equations (4.4) and (4.5) as a system of first order PDEs by introducing the diffusive fluxes (Nguyen et al., 2009a,b; Kirby et al., 2012):

$$\mathbf{q}_s = -\lambda_o \mathbf{K} \nabla p_c, \qquad \mathbf{q}_p = -\lambda_t \mathbf{K} \nabla p_w.$$

Thus, the saturation system is

$$\begin{cases} \phi \dfrac{\partial S_o}{\partial t} + \nabla \cdot \left(\mathbf{q}_s + \dfrac{\lambda_o}{\lambda_t} \mathbf{q}_p \right) = f_o(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ \mathbf{q}_s + \lambda_o \mathbf{K} \nabla p_c = \mathbf{0} & \forall (\mathbf{x}, t) \in (\Omega, T), \\ S_o^{\Gamma_{in}}(\mathbf{x}, t) = g_{Ds}^{in}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{in}, T), \\ \left(\dfrac{\lambda_o \lambda_w}{\lambda_t} \mathbf{K} \nabla p_c \right) \cdot \mathbf{n} = g_{Ns}^{out}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{out}, T), \\ \mathbf{v}_o \cdot \mathbf{n} = 0 & \forall (\mathbf{x}, t) \in (\Gamma_{nf}, T), \\ S_o(\cdot, 0) = S_o^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega, \end{cases} \tag{4.6}$$

and the pressure system is

$$\begin{cases} \nabla \cdot (\mathbf{q}_p + \mathbf{q}_s) = f_o(\mathbf{x}, t) + f_w(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Omega, T), \\ \mathbf{q}_p + \lambda_t \mathbf{K} \nabla p_w = \mathbf{0} & \forall (\mathbf{x}, t) \in (\Omega, T), \\ p_w^{\Gamma_{in}}(\mathbf{x}, t) = g_{Dp}^{in}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{in}, T), \\ p_w^{\Gamma_{out}}(\mathbf{x}, t) = g_{Dp}^{out}(\mathbf{x}, t) & \forall (\mathbf{x}, t) \in (\Gamma_{out}, T), \\ \mathbf{v}_t \cdot \mathbf{n} = 0 & \forall (\mathbf{x}, t) \in (\Gamma_{nf}, T). \end{cases} \tag{4.7}$$

4.3 Spatial discretization

We discretize the domain, Ω, with a tessellation, T_h, composed of a set of elements, e, of polynomial degree P. Afterwards, we introduce the discontinuous finite element

spaces associated with the tessellation, T_h:

$$\begin{aligned}
\mathcal{V}_h^P &= \left\{ v \in L^2\left(\Omega^d\right) \;\middle|\; v_{|e} \in \left(\mathbb{S}^P\left(e\right)\right) \forall e \in \mathrm{T}_h \right\}, \\
\mathcal{W}_h^P &= \left\{ \mathbf{w} \in \left(L^2\left(\Omega^d\right)\right)^d \;\middle|\; \mathbf{w}_{|e} \in \left(\mathbb{S}^P\left(e\right)\right)^d \forall e \in \mathrm{T}_h \right\}, \\
\mathcal{M}_h^P &= \left\{ \gamma \in L^2\left(\Sigma_h\right) \;\middle|\; \gamma_{|f} \in \left(\mathbb{S}^P\left(f\right)\right) \forall f \in \Sigma_h \right\},
\end{aligned}$$

where $\mathbb{S}^P$ is the space of the polynomials of degree at most P for triangles and tetrahedra (usually denoted by $\mathbb{P}^P$), or the tensor products of polynomials of degree at most P in each coordinate direction for tensor product elements (usually denoted by $\mathbb{Q}^P$), d is the space dimension and Σ_h is the skeleton of the mesh composed of all the element faces, f. We define $\mathcal{M}_h^P\left(g_D\right) = \left\{ \gamma \in \mathcal{M}_h^P \;\middle|\; \gamma = \Pi(g_D) \text{ on } \Gamma_D \right\}$, where $\Pi(\cdot)$ is a projection operator to the space $\left\{ \gamma|_{\Gamma_D} \, \forall \gamma \in \mathcal{M}_h^P \right\}$, and g_D is the prescribed values for the saturation and pressure equations on their respective Dirichlet boundaries, Γ_D. Note that, for the saturation equation the Dirichlet boundary is defined on the inflow boundary, and for the pressure is defined on the inflow and outflow boundaries. In this work, we use a fixed polynomial degree for all the elements. We define the scalar products on the finite element spaces:

$$\begin{aligned}
\left(u, v\right)_e &= \int_e u \; v \; \mathrm{d}\Omega \qquad && \forall u, v \in \mathcal{V}_h^P, \\
\left(\mathbf{q}, \mathbf{w}\right)_e &= \int_e \mathbf{q} \; \cdot \mathbf{w} \; \mathrm{d}\Omega \qquad && \forall \mathbf{q}, \mathbf{w} \in \mathcal{W}_h^P, \\
\left\langle \lambda, \gamma \right\rangle_{\partial e} &= \int_{\partial e} \lambda \; \gamma \; \mathrm{d}\Gamma \qquad && \forall \lambda, \gamma \in \mathcal{M}_h^P.
\end{aligned}$$

The HDG formulation for the oil saturation corresponding to Equation (4.6) seeks an approximation $(S_{o_h}, \mathbf{q}_{s_h}, \hat{S}_{oh}) \in \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(g_{Ds}^{in})$ such that:

$$\begin{aligned}
&\sum_{e \in \mathrm{T}_h} \left((\phi \frac{\partial S_{o_h}}{\partial t}, v)_e - (\mathbf{q}_{s_h} + \frac{\lambda_o}{\lambda_t} \mathbf{q}_{p_h}, \nabla v)_e \right) \\
&+ \sum_{e \in \mathrm{T}_h} \left(\langle (\hat{\mathbf{q}}_{s_h} + \frac{\hat{\lambda}_o}{\hat{\lambda}_t} \hat{\mathbf{q}}_{p_h}) \cdot \mathbf{n}, v \rangle_{\partial e} - (f_o, v)_e \right) = 0,
\end{aligned} \tag{4.8a}$$

$$\sum_{e \in \mathrm{T}_h} \left((\mathbf{A}_{s_h}^{-1} \mathbf{q}_{s_h}, \mathbf{w})_e - (S_{o_h}, \nabla \cdot \mathbf{w})_e + \langle \hat{S}_{o_h}, \mathbf{w} \cdot \mathbf{n} \rangle_{\partial e} \right) = 0, \tag{4.8b}$$

$$\sum_{e \in \mathrm{T}_h} \left(\langle (\hat{\mathbf{q}}_{s_h} + \frac{\hat{\lambda}_o}{\hat{\lambda}_t} \hat{\mathbf{q}}_{p_h}) \cdot \mathbf{n}, \gamma \rangle_{\partial e} \right) - \langle g_{Ns}^{out}, \gamma \rangle_{\Gamma_{out}} = 0, \tag{4.8c}$$

for all $(v, \mathbf{w}, \gamma) \in \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(0)$, where $\hat{\lambda}_o$ and $\hat{\lambda}_t$ are the oil phase mobility and the total phase mobility computed with the trace of the oil saturation, $\hat{S}_{o_h}$,

respectively, and $\mathbf{A}_{s_h} = \lambda_o p_c' \mathbf{K}$, being p_c' the derivative of the capillary pressure respect to the oil saturation.

The HDG formulation for the water pressure corresponding to Equation (4.7) seeks an approximation $(p_{w_h}, \mathbf{q}_{p_h}, \hat{p}_{w_h}) \in \mathcal{V}_h^P \times \mathcal{W}_h^P \times \mathcal{M}_h^P(g_{Dp})$ such that:

$$\sum_{e \in \mathrm{T}_h} \left(-(\mathbf{q}_{p_h} + \mathbf{q}_{s_h}, \nabla v)_e + \langle (\hat{\mathbf{q}}_{p_h} + \hat{\mathbf{q}}_{s_h}) \cdot \mathbf{n}, v \rangle_{\partial e} - (f_o + f_w, v)_e \right) = 0, \tag{4.9a}$$

$$\sum_{e \in \mathrm{T}_h} \left((\mathbf{A}_{p_h}^{-1} \mathbf{q}_{p_h}, \mathbf{w})_e - (p_{w_h}, \nabla \cdot \mathbf{w})_e + \langle \hat{p}_{w_h}, \mathbf{w} \cdot \mathbf{n} \rangle_{\partial e} \right) = 0, \tag{4.9b}$$

$$\sum_{e \in \mathrm{T}_h} \langle (\hat{\mathbf{q}}_{p_h} + \hat{\mathbf{q}}_{s_h}) \cdot \mathbf{n}, \gamma \rangle_{\partial e} = 0, \tag{4.9c}$$

for all $(v, \mathbf{w}, \gamma) \in \mathcal{W}_h^P \times \mathcal{V}_h^P \times \mathcal{M}_h^P(0)$, where g_{Dp} are the Dirichlet conditions for the pressure, $\mathbf{A}_{p_h} = \lambda_t \mathbf{K}$, and $\hat{p}_{w_h}$ is the trace of the water pressure.

Equations (4.8c) and (4.9c) are the transmissivity equations, in which we impose the continuity of the total numerical flux in the normal direction between adjacent elements for each equation. Therefore, these equations relate the unknowns between adjacent elements. We define the numerical flux for the oil saturation and the water pressure as

$$\hat{\mathbf{q}}_{s_h} = \mathbf{q}_{s_h} + \tau_s (S_{o_h} - \hat{S}_{o_h}) \mathbf{n}, \tag{4.10a}$$

$$\hat{\mathbf{q}}_{p_h} = \mathbf{q}_{p_h} + \tau_p (p_{w_h} - \hat{p}_{w_h}) \mathbf{n}, \tag{4.10b}$$

respectively, where τ_s is the stabilization function for the oil saturation, and τ_p is the stabilization function for the water pressure. According to Nguyen et al. (2009a,b), we set the stabilization parameters, τ_s and τ_p, of Equations(4.10a) and (4.10b) respectively as

$$\tau_s = \frac{\hat{\lambda}_o p_c'}{l_s} \gamma_{\mathbf{K}}, \qquad \tau_p = \frac{\hat{\lambda}_t}{l_p} \gamma_{\mathbf{K}}, \tag{4.11}$$

where $\gamma_{\mathbf{K}}$ is the maximum eigenvalue of the permeability matrix, $\mathbf{K}$, l_s is the characteristic length for the saturation and l_p is the characteristic length for the pressure.

We highlight that the Dirichlet boundary conditions are applied as follows:

$$\hat{S}_{o_h} = \Pi(g_{Ds}^{in}) \quad \forall \mathbf{x} \in \partial \mathrm{T}_{h\Gamma_D}^s, \qquad \hat{p}_{w_h} = \Pi(g_{Dp}) \quad \forall \mathbf{x} \in \partial \mathrm{T}_{h\Gamma_D}^p,$$

where $\partial \mathrm{T}_{h\Gamma_D}^s$ and $\partial \mathrm{T}_{h\Gamma_D}^p$ are the set of mesh faces on the Dirichlet boundary for the oil saturation and water pressure, respectively.

Let $\{N_i\}_{i=1,\dots,N}$ be a Lagrangian basis of shape functions of $\mathbb{S}^P(e)$, where N is the total number of element nodes, and let $\{N_i^f\}_{i=1,\dots,N_f}$ be a Lagrangian basis on the element faces $\mathbb{S}^P(f)$, where N_f is the total number of nodes on the element faces. Thus, S_{o_h}, $\mathbf{q}_{s_h}$, $\hat{S}_{o_h}$, p_{w_h}, $\mathbf{q}_{p_h}$ and $\hat{p}_{w_h}$are defined as

$$
\begin{aligned}
S_{o_h}(\mathbf{x},t) &= \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N} S_i(t)N_i(\mathbf{x}), & \mathbf{q}_{s_h}(\mathbf{x},t) &= \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N}\sum_{j=1}^{N_{sd}} q_{si,j}(t)N_i(\mathbf{x})\mathbf{e}_j,\\
\hat{S}_{o_h}(\mathbf{x},t) &= \sum_{f\in \Sigma_h}\sum_{i=1}^{N_f} \hat{S}_l(t)N_l^f(\mathbf{x}), & p_{w_h}(\mathbf{x},t) &= \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N} p_i(t)N_i(\mathbf{x}),\\
\mathbf{q}_{p_h}(\mathbf{x},t) &= \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N}\sum_{j=1}^{N_{sd}} q_{p_{i,j}}(t)N_i(\mathbf{x})\mathbf{e}_j, & \hat{p}_{w_h}(\mathbf{x},t) &= \sum_{f\in \Sigma_h}\sum_{i=1}^{N_f} \hat{p}_l(t)N_l^f(\mathbf{x}).
\end{aligned}
\tag{4.12}
$$

Similarly, the temporal derivative of the oil saturation, $\dot{S}_{o_h} = \partial S_{o_h}/\partial t$, is defined as

$$
\dot{S}_{o_h}(\mathbf{x},t) = \sum_{e\in \mathrm{T}_h}\sum_{i=1}^{N} \dot{S}_i(t)N_i(\mathbf{x}), \tag{4.13}
$$

where $\dot{S}_i(t) = dS_i(t)/dt$. By inserting Equations (4.12) and (4.13) into Equations (4.8) and (4.9), we obtain a coupled system of first order differential algebraic equations (DAE). Specifically, the discrete problem consists of finding the coefficients $S_i(t)$, $\dot{S}_i(t)$, $q_{si,j}(t)$, $p_i(t)$, $q_{p_{i,j}}(t)$, for $i = 1\dots N$, $j = 1\dots N_{sd}$, and $\hat{S}_l(t), \hat{p}_l(t)$ for $l = 1\dots N_f$ such that

$$
\begin{aligned}
\left[\mathbf{R}_{S_o}\right]_i \equiv& \sum_{e\in \mathrm{T}_h}\left(\left(\phi\dot{S}_{o_h}, N_i\right)_e - \left(\mathbf{q}_{s_h} + \frac{\lambda_o}{\lambda_t}\mathbf{q}_{p_h}, \nabla N_i\right)_e\right)\\
&+ \sum_{e\in \mathrm{T}_h}\left(\left\langle \mathbf{q}_{s_h}\cdot\mathbf{n} + \tau_s(S_{o_h} - \hat{S}_{o_h}), N_i\right\rangle_{\partial e}\right)\\
&+ \sum_{e\in \mathrm{T}_h}\left(\left\langle \frac{\hat{\lambda}_o}{\hat{\lambda}_t}\left(\mathbf{q}_{p_h}\cdot\mathbf{n} + \tau_p(p_{w_h} - \hat{p}_{w_h})\right), N_i\right\rangle_{\partial e}\right) - \sum_{e\in \mathrm{T}_h}(f_o, N_i)_e = 0,\\
\left[\mathbf{R}_{\mathbf{q}_s}\right]_{i,j} \equiv& \sum_{e\in \mathrm{T}_h}\left((\mathbf{A}_{s_h}^{-1}\mathbf{q}_{s_h}, N_i\mathbf{e}_j)_e - (S_{o_h}, \nabla\cdot(N_i\mathbf{e}_j))_e + \langle \hat{S}_{o_h}, N_i\mathbf{e}_j\cdot\mathbf{n}\rangle_{\partial e}\right) = 0,\\
\left[\mathbf{R}_{\hat{S}_o}\right]_l \equiv& \sum_{e\in \mathrm{T}_h}\left(\left\langle \mathbf{q}_{s_h}\cdot\mathbf{n} + \tau_s(S_{o_h} - \hat{S}_{o_h}) + \frac{\hat{\lambda}_o}{\hat{\lambda}_t}(\mathbf{q}_{p_h}\cdot\mathbf{n} + \tau_p(p_{w_h} - \hat{p}_{w_h})), N_l^f\right\rangle_{\partial e}\right)\\
&- \langle g_{Ns}^{out}, N_l^f\rangle_{\Gamma_{out}} = 0,
\end{aligned}
$$

$$\begin{aligned}
\left[\mathbf{R}_{p_w}\right]_i &\equiv \sum_{e\in \mathrm{T}_h} \left(-(\mathbf{q}_{p_h}+\mathbf{q}_{s_h}, \nabla N_i)_e + \langle \mathbf{q}_{p_h}\cdot\mathbf{n} + \tau_p(p_{w_h}-\hat{p}_{w_h}), N_i\rangle_{\partial e}\right) \\
&\quad + \sum_{e\in \mathrm{T}_h} \left(\langle \mathbf{q}_{s_h}\cdot\mathbf{n} + \tau_s(S_{o_h}-\hat{S}_{o_h}), N_i\rangle_{\partial e}\right) - \sum_{e\in \mathrm{T}_h} (f_o+f_w, N_i)_e = 0, \\
\left[\mathbf{R}_{\mathbf{q}_p}\right]_{i,j} &\equiv \sum_{e\in \mathrm{T}_h} \left((\mathbf{A}_{p_h}^{-1}\mathbf{q}_{p_h}, N_i\mathbf{e}_j)_e - (p_{w_h}, \nabla\cdot(N_i\mathbf{e}_j))_e + \langle \hat{p}_{w_h}, N_i\mathbf{e}_j\cdot\mathbf{n}\rangle_{\partial e}\right) = 0, \\
\left[\mathbf{R}_{\hat{p}_w}\right]_l &\equiv \sum_{e\in \mathrm{T}_h} \left(\langle \mathbf{q}_{p_h}\cdot\mathbf{n} + \tau_p(p_{w_h}-\hat{p}_{w_h}) + \mathbf{q}_{s_h}\cdot\mathbf{n} + \tau_s(S_{o_h}-\hat{S}_{o_h}), N_l^f\rangle_{\partial e}\right) = 0,
\end{aligned}$$

for $N_i, N_i\mathbf{e}_j$ and N_l^f, with $i = 1\dots N$, $j = 1\dots N_{sd}$, $l = 1\dots N_f$.

4.4 Temporal discretization

We rewrite the DAE system in a compact form as:

$$\mathbf{R}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) = \begin{bmatrix}
\mathbf{R}_{S_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\
\mathbf{R}_{\mathbf{q}_s}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\
\mathbf{R}_{\hat{S}_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\
\mathbf{R}_{p_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\
\mathbf{R}_{\mathbf{q}_p}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\
\mathbf{R}_{\hat{p}_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right)
\end{bmatrix} = \mathbf{0}, \quad (4.14)$$

where $\mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p$ and $\hat{\mathbf{p}}_w$ are vectors composed of all the nodal values for the oil saturation, $S_i(t)$, the derivative of the oil saturation, $\dot{S}_i(t)$, the numerical flux for the oil saturation, $q_{si,j}(t)$, the trace of the oil saturation, $\hat{S}_i(t)$, the water pressure, $p_i(t)$, the numerical flux for the water pressure, $q_{p_{i,j}}(t)$, and the traces of the water pressure, $\hat{p}_i(t)$.

To solve the DAE in Equation (4.14), we use a diagonally implicit Runge-Kutta method (DIRK). From now on, we denote by $(\cdot)^n$ the value of any variable at time t^n and by $(\cdot)^{n,i}$ the value of any variable at time $t^{n,i} = t^n + c^i\Delta t$, being n the time step and i the DIRK stage. Accordingly, we compute the oil saturation at time $t^{n+1} = t^n + \Delta t$ as:

$$\mathbf{S}_o^{n+1} = \mathbf{S}_o^n + \Delta t \sum_{i=1}^{s} b^i \dot{\mathbf{S}}_o^{n,i},$$

where $\dot{\mathbf{S}}_o^{n,i}$ is the approximation of $\dot{\mathbf{S}}_o$ at time $t^{n,i}$, and s is the total number of stages. The oil saturation at each stage of the DIRK scheme is computed as:

$$\mathbf{S}_o^{n,i} = \mathbf{S}_o^n + \Delta t \sum_{j=1}^{i} a^{ij} \dot{\mathbf{S}}_o^{n,j}, \tag{4.15}$$

and the $\dot{\mathbf{S}}_o^{n,i}$ for $i = 1, \ldots, s$ are computed as the solution of the non-linear algebraic equation:

$$\mathbf{R}\left(t^{n,i}, \mathbf{S}_o^n + \Delta t \sum_{j=1}^{i} a^{ij} \dot{\mathbf{S}}_o^{n,j}, \dot{\mathbf{S}}_o^{n,i}, \mathbf{q}_s^{n,i}, \hat{\mathbf{S}}_o^{n,i}, \mathbf{p}_w^{n,i}, \mathbf{q}_p^{n,i}, \hat{\mathbf{p}}_w^{n,i}\right) = \mathbf{0}. \tag{4.16}$$

The parameters a^{ij}, b^i, c^i, with $i = 1 \ldots s$ and $j = 1 \ldots i$, define the DIRK method and are given by the Butcher's tables (Butcher, 1964a,b, 2016; Montlaur et al., 2012; Kennedy and Carpenter, 2016; Pazner and Persson, 2017), see Equation (3.21) in Chapter 3.

4.5 Non-linear solver

To solve Equation (4.16), we use a fix-point iteration method. The main idea is to iteratively solve the saturation and the pressure until convergence is achieved, see Algorithm 4.1. To this aim, let l be the l-th iteration of the non-linear solver. Thus, we first solve Equation (4.16) for the oil saturation by imposing:

$$\begin{aligned}\mathbf{R}\Bigg(&\underbrace{t^{n,i}, \mathbf{S}_o^n + \Delta t \sum_{j=1}^{i-1} a^{ij} \dot{\mathbf{S}}_o^{n,j}}_{\text{Data}} + \underbrace{\Delta t a^{ii} \dot{\mathbf{S}}_o^{n,i,l+1}, \dot{\mathbf{S}}_o^{n,i,l+1}, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{\text{Unknowns}}, \\ &\underbrace{\mathbf{p}_w^{n,i,l}, \mathbf{q}_p^{n,i,l}, \hat{\mathbf{p}}_w^{n,i,l}}_{\text{Data}}\Bigg) = \mathbf{0},\end{aligned} \tag{4.17}$$

from which we compute $(\dot{\mathbf{S}}_o^{n,i,l+1}, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1})$ given $(\mathbf{p}_w^{n,i,l}, \mathbf{q}_p^{n,i,l}, \hat{\mathbf{p}}_w^{n,i,l})$, Line 6 of Algorithm 4.1. We compute $\mathbf{S}_o^{n,i,l+1}$ using Equation (4.15), Line 7 of Algorithm 4.1. Then, we also solve Equation (4.16) for the water pressure by imposing:

$$\begin{aligned}\mathbf{R}\Bigg(&\underbrace{t^{n,i}, \mathbf{S}_o^n + \Delta t \sum_{j=1}^{i-1} a^{ij} \dot{\mathbf{S}}_o^{n,j} + \Delta t a^{ii} \dot{\mathbf{S}}_o^{n,i,l+1}, \dot{\mathbf{S}}_o^{n,i,l+1}, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{\text{Data}}, \\ &\underbrace{\mathbf{p}_w^{n,i,l+1}, \mathbf{q}_p^{n,i,l+1}, \hat{\mathbf{p}}_w^{n,i,l+1}}_{\text{Unknowns}}\Bigg) = \mathbf{0},\end{aligned} \tag{4.18}$$

Algorithm 4.1 Fix-point iteration method for two-phase flow at each DIRK stage.

1: **Input**: $\mathbf{S}_o^n$
2: $\mathbf{S}_o^{n,0} = \mathbf{S}_o^{n-1}$, $\mathbf{p}_w^{n,0} = \mathbf{p}_w^{n-1}$.
3: $\mathrm{l} = 0$
4: convergence = False
5: **While** (not convergence):
6: **Compute**: $\dot{\mathbf{S}}_o^{n,i,l+1}$, $\mathbf{q}_s^{n,i,l+1}$ and $\hat{\mathbf{S}}_o^{n,i,l+1}$ from $\mathbf{S}_o^{n-1}$, $\mathbf{p}_w^{n,l}$, $\mathbf{q}_p^{n,i,l}$ and $\hat{\mathbf{p}}_w^{n,i,l}$ using Eq.(4.17).
7: **Compute**: $\mathbf{S}_o^{n,i,l+1}$ using Eq.(4.15).
8: **Compute**: $\mathbf{p}_w^{n,i,l+1}$, $\mathbf{q}_p^{n,i,l+1}$ and $\hat{\mathbf{p}}_w^{n,i,l+1}$ from $\mathbf{S}_o^{n,l+1}$, $\mathbf{q}_s^{n,i,l+1}$ and $\hat{\mathbf{S}}_o^{n,i,l+1}$ using Eq.(4.18).
9: $l = l + 1$
10: **check convergence** using Eq.(4.19).
11: $\mathbf{S}_o^{n,i} = \mathbf{S}_o^{n,i,l}$, $\mathbf{q}_s^{n,i} = \mathbf{q}_s^{n,i,l}$, $\hat{\mathbf{S}}_o^{n,i} = \hat{\mathbf{S}}_o^{n,i,l}$, $\mathbf{p}_w^{n,i} = \mathbf{p}_w^{n,i,l}$, $\mathbf{q}_p^{n,i} = \mathbf{q}_p^{n,i,l}$, $\hat{\mathbf{p}}_w^{n,i} = \hat{\mathbf{p}}_w^{n,i,l}$.
12: end

from which we obtain $(\mathbf{p}_w^{n,i,l+1}, \mathbf{q}_p^{n,i,l+1}, \hat{\mathbf{p}}_w^{n,i,l+1})$ given $(\dot{\mathbf{S}}_o^{n,i,l+1}, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1})$, Line 8 of Algorithm 4.1.

This procedure will be repeated until convergence is achieved at each Runge-Kutta stage, $i = 1, \ldots, s$, Line 10 of Algorithm 4.1. We define the stopping criteria of the non-linear solver using appropriate tolerances as

$$
\begin{aligned}
&\frac{\|S_{o_h}^{n,i,l} - S_{o_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|S_{o_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} < \varepsilon_{S_o}, && \frac{\|p_{w_h}^{n,i,l} - p_{w_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|p_{w_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} < \varepsilon_{p_w},\\
&\frac{\|\mathbf{q}_{s_h}^{n,i,l} - \mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|\mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} < \varepsilon_{\mathbf{q}_s}, && \frac{\|\mathbf{q}_{p_h}^{n,i,l} - \mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|\mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} < \varepsilon_{\mathbf{q}_p},\\
&\frac{\|\hat{S}_{o_h}^{n,i,l} - \hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}}{\|\hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}} < \varepsilon_{\hat{S}_o}, && \frac{\|\hat{p}_{w_h}^{n,i,l} - \hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}}{\|\hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}} < \varepsilon_{\hat{p}_w},\\
&\|\mathbf{R}_{S_o}\|_2 < \varepsilon_{\mathbf{R}_{S_o}}, && \|\mathbf{R}_{p_w}\|_2 < \varepsilon_{\mathbf{R}_{p_w}},\\
&\|\mathbf{R}_{\mathbf{q}_s}\|_2 < \varepsilon_{\mathbf{R}_{\mathbf{q}_s}}, && \|\mathbf{R}_{\mathbf{q}_p}\|_2 < \varepsilon_{\mathbf{R}_{\mathbf{q}_p}},\\
&\|\mathbf{R}_{\hat{S}_o}\|_2 < \varepsilon_{\mathbf{R}_{\hat{S}_o}}, && \|\mathbf{R}_{\hat{p}_w}\|_2 < \varepsilon_{\mathbf{R}_{\hat{p}_w}},
\end{aligned}
\tag{4.19}
$$

where $\|\cdot\|_{L^2(\mathrm{T}_h)}$ is the norm of the $L^2(\mathrm{T}_h)$ space of functions, $\|\cdot\|_{L^2(\Sigma_h)}$ is the norm of the $L^2(\Sigma_h)$ space of functions, and $\|\cdot\|_2$ is the Euclidean norm of vectors.

We highlight that for each iteration of the fix-point method we need to solve two linear systems, one for the saturation, Equation (4.17) and another for the pressure, Equation (4.18). Each system to be solved is hybridized in order to solve a linear system for $\hat{\mathbf{S}}_o^{n,i,l+1}$ and another for $\hat{\mathbf{p}}_w^{n,i,l+1}$. The other unknowns are recovered using

an element-wise process. The hybridization procedure is applied in the saturation and pressure systems similarly as in Section 3.6 of Chapter 3.

The proposed method differs from the classical and widely used IMPES method to solve two-phase flow problem. IMPES method solves the pressure implicitly and the saturation explicitly, whereas our formulation solves both variables implicitly (Chen et al., 2006). The implicit scheme allows using larger time steps. In addition it is high-order accurate in time. Moreover, the method detailed in Algorithm 4.1 iterates until convergence is achieved at each stage delivering more accurate approximations.

4.6 Local post-processing

One of the main advantages of using the HDG formulation is that the scalar variables (pressure and saturation) and their fluxes, in $\mathcal{V}_h^P$ and $\mathcal{W}_h^P$ spaces respectively, have a rate of convergence of $P+1$ in the L^2-norm, when the temporal error is low enough. Moreover, a local post-processing can be applied to obtain a new approximation for the saturation, $S_{o_h}^*$, and for the pressure, $p_{w_h}^*$, both in $\mathcal{V}_h^{P+1}$ with convergence rate of $P+2$ in the L^2-norm (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018).

In our formulation, we have two local problems. The first one consists of finding the post-processed saturation, $S_{o_h}^* \in \mathcal{V}_h^{P+1}$ on each element, e, such that:

$$\begin{aligned}(\mathbf{K}\lambda_o p_c' \nabla S_{o_h}^*, \nabla v)_e = & -(\mathbf{q}_{s_h}, \nabla v)_e \\ (S_{o_h}^*, 1)_e = & (S_{o_h}, 1)_e,\end{aligned} \tag{4.20}$$

for all $v \in \mathcal{V}_h^{P+1}$. The second local problem consists of finding the post-processed pressure, $p_{w_h}^* \in \mathcal{V}_h^{P+1}$ on each element, e, such that:

$$\begin{aligned}(\mathbf{K}\lambda_t \nabla p_{w_h}^*, \nabla v)_e = & -(\mathbf{q}_{p_h}, \nabla v)_e \\ (p_{w_h}^*, 1)_e = & (p_{w_h}, 1)_e,\end{aligned} \tag{4.21}$$

for all $v \in \mathcal{V}_h^{P+1}$.

It is important to highlight that this procedure can be applied at selected time steps, and it is not necessary to apply it at all the time steps (Kirby et al., 2012; Nguyen et al., 2009a,b).

4.7 Examples

In this section, we first validate the proposed formulation and the applied methodology by showing numerical evidence of the convergence rates. Second, we apply our method to simulate two applications of waterflooding technique for oil recovery. Specifically, in the second example, we solve the classical five spot pattern problem, and in the third one, we consider an heterogeneous domain.

In all the examples, we use the DIRK3-s3 scheme defined in Appendix B to perform the time discretization. All the high-order meshes are generated using the algorithms presented in Gargallo-Peiró et al. (2015, 2016); Ruiz-Gironés E.; Roca and Sarrate (2016), which are implemented in the EZ4U environment (Roca et al., 2010).

In the last two examples, we use the Brooks-Corey model, Equation (4.1), with $p_e = 10^3\,\text{Pa}$, $\theta = 2$, $S_{rw} = 0.2$ and $S_{ro} = 0.2$. In those examples, we define the following boundary conditions and source terms:

$$\begin{array}{llllll} p_w^{\Gamma_{in}} & = 3\cdot 10^6 \text{ Pa}, & S_o^{\Gamma_{in}} & = g_{Ds}^{in}, & \text{on } \Gamma_{in}, \\ p_w^{\Gamma_{out}} & = 10^6 \text{ Pa}, & \left(\dfrac{\lambda_o\lambda_w}{\lambda_t}\mathbf{K}\nabla p_c\right)\cdot\mathbf{n} & = 0, & \text{on } \Gamma_{out}, \\ \mathbf{v}_t\cdot\mathbf{n} & = 0, & \mathbf{v}_o\cdot\mathbf{n} & = 0, & \text{on } \Gamma_{n_f}, \\ f_w & = 0, & f_o & = 0, & \text{in } \Omega, \end{array} \tag{4.22}$$

where g_{Ds}^{in} will be different in each example. The initial condition for the saturation is $S_o(\cdot,0) = 0.78$. We set the characteristic lengths in Equation (4.11) as $l_s = 10^{-7}$ and $l_p = 10^{-10}$. In addition, we define the tolerances of the non-linear solver, Equation (4.19), as:

$$\begin{aligned} &\varepsilon_{S_o} = 10^{-8}, \varepsilon_{\mathbf{q}_s} = 10^{-6}, \quad \varepsilon_{\hat{S}_o} = 10^{-8}, \varepsilon_{\mathbf{R}_{S_o}} = 10^{-8}, \quad \varepsilon_{\mathbf{R}_{\mathbf{q}_s}} = 10^{-6}, \varepsilon_{\mathbf{R}_{\hat{S}_o}} = 10^{-8}, \\ &\varepsilon_{p_w} = 10^{-8}, \varepsilon_{\mathbf{q}_p} = 10^{-6}, \quad \varepsilon_{\hat{p}_w} = 10^{-8}, \varepsilon_{\mathbf{R}_{p_w}} = 10^{-8}, \quad \varepsilon_{\mathbf{R}_{\mathbf{q}_p}} = 10^{-6}, \varepsilon_{\mathbf{R}_{\hat{p}_w}} = 10^{-8}. \end{aligned}$$

4.7.1 Convergence rate analysis

In this example, we show numerical evidence of the convergence rates for the S_{o_h}, p_{w_h}, $\mathbf{q}_{s_h}$, $\mathbf{q}_{p_h}$, and the post-processed saturation and pressure, $S_{o_h}^*$ and p_h^*, respectively, of our formulation. This example serves as a validation of our formulation and the proposed methodology.

We define an analytical solution for the oil saturation, S_o, and the water pressure, p_w, as:

$$S_o = 0.5 + t\left(\frac{\sin(\pi x)\sin(\pi y)}{4}\right), \qquad p_w = 0.5 + t\left(\frac{\cos(\pi x)\cos(\pi y)}{4}\right),$$

where $(x, y) \in \Omega = [0, 1]$ m$\times[0, 1]$ m and $t \in (0, 1)$ s. We prescribe Dirichlet boundary condition for the water pressure and oil saturation, and the source terms are taken to satisfy the PDEs with the proposed analytical solutions. We set the soil permeability as $\mathbf{K} = \mathbf{I}$ m^2, the porosity as $\phi = 0.1$, the oil viscosity as $\mu_o = 1$ Pa $\cdot$ s and the water viscosity as $\mu_w = 0.1$ Pa $\cdot$ s. Specifically for this example, we use the Brooks-Corey model, Equation (4.1), with $p_e = 0.5$ Pa, $\theta = 1$. We select $\tau_p = 1$ and $\tau_s = 10$, see Equation(4.11). Specifically for this example, we set to 10^{-10} all the tolerances of the fix-point solver, see Equation (4.19).

To study the convergence rates, we discretize Ω using quadrilateral meshes composed between 9 to 6400 elements of polynomial degree from 1 to 3. Moreover, we use a DIRK3-s3 scheme with $\Delta t = 0.25$ s for the time discretization. At $t = 1$ s, we measure the error in the L^2-norm of the numerical solutions against the analytical ones.

Figure 4.1 shows the convergence rates of the saturation and pressure, their fluxes, and the post-processed solutions. We obtain the expected convergence rate of $P + 1$ in L^2-norm for the water pressure, the oil saturation and for the fluxes $\mathbf{q}_p$ and $\mathbf{q}_s$. The local post-process, detailed in Equations (4.20) and (4.21), is applied to obtain a super convergence rate of $P + 2$ in L^2-norm of the post-processed pressure, p_h^*, and the post-processed saturation, $S_{o_h}^*$.

This example shows two main advantages of using HDG formulation, which are the optimal convergence rate of $P + 1$ in L^2-norm for the solutions and their fluxes, and the convergence rate of $P + 2$ in L^2-norm for the post-processed solutions.

4.7.2 Five spot pattern

We consider a square domain, Ω, of 140 m $\times$ 140 m. The selected pattern has four injection wells located at the vertices of square, and one producer well at its center, see Figure 4.2(a). The radius of the wells is $r_w = 5$ m. On the boundary $\Gamma = \Gamma_{in} \bigcup \Gamma_{out} \bigcup \Gamma_{nf}$, we apply the boundary conditions detailed in Equation (4.22), with $g_{Ds}^{in} = 0.3$.

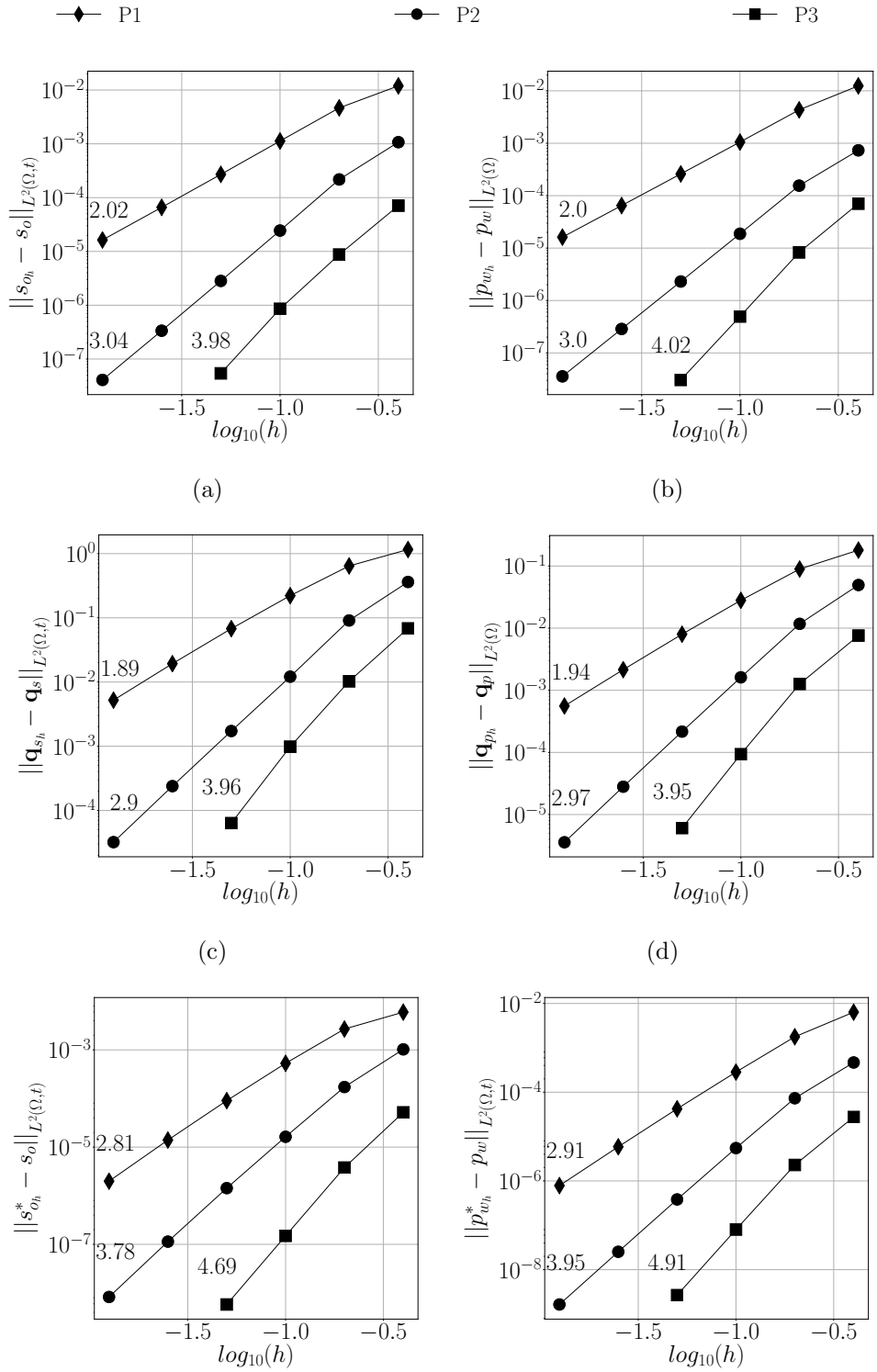
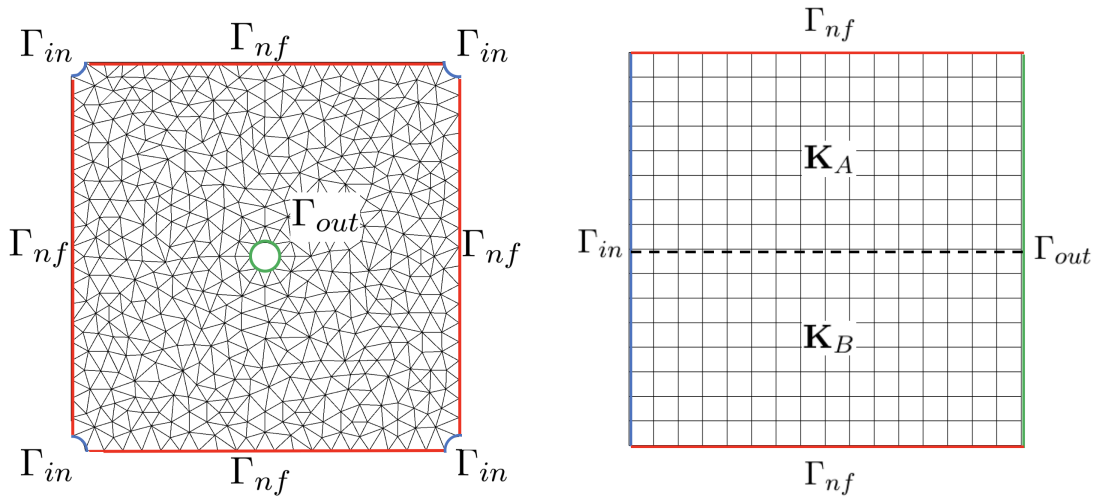


Figure 4.1: Convergence rates for: a) saturation, b) pressure, c) saturation flux, d) pressure flux; e) post-processed saturation, and f) post-processed pressure.



(a) (b)

Figure 4.2: Mesh and boundary conditions distributions for the examples of the waterflooding simulation through: a) the five spot pattern b) the heterogeneous domain.

The soil permeability is $\mathbf{K} = 10^{-11}\mathbf{I}$ m^2, the porosity is $\phi = 0.2$, and the viscosity for the water and oil phases are $\mu_w = 0.001$ Pa $\cdot$ s and $\mu_o = 0.01$ Pa $\cdot$ s, respectively.

We discretize Ω with 930 triangular elements of polynomial degree three (4314 nodes), see Figure 4.2(a). The number of unknowns involved in the linear systems that have to be solved at each iteration of Equations (4.17) and (4.18) is 33652. Nevertheless, applying a static condensation procedure in the HDG formulation it is reduced to 5752 unknowns for each one. The time step for the DIRK3-s3 scheme is $\Delta t = 900$ s.

Figure 4.3 shows the water pressure and saturation approximations at time $t = 51$ h and $t = 65$ h. At the initial stage the water saturation, S_w, is equal to the residual water saturation, $S_w(\cdot, 0) = 1 - S_o(\cdot, 0)$, and oil occupies the rest of the voids. Afterwards, water is injected from the corner wells (injectors), moving the oil to the pumping well at the center, and occupies the space left by the oil. This is observed in Figures 4.3(a) and 4.3(b), in which we show the water saturation and the water velocity vectors at time 51 and 65 hours, respectively. The water saturation increases from the injectors wells to the producer well, as the result of mobilising the oil to the producer well. Moreover, the water moves away from injector wells, while it moves towards the producer well. Figures 4.3(c) and 4.3(d), show the water pressure field. As expected, the water pressure has higher values at the injector wells

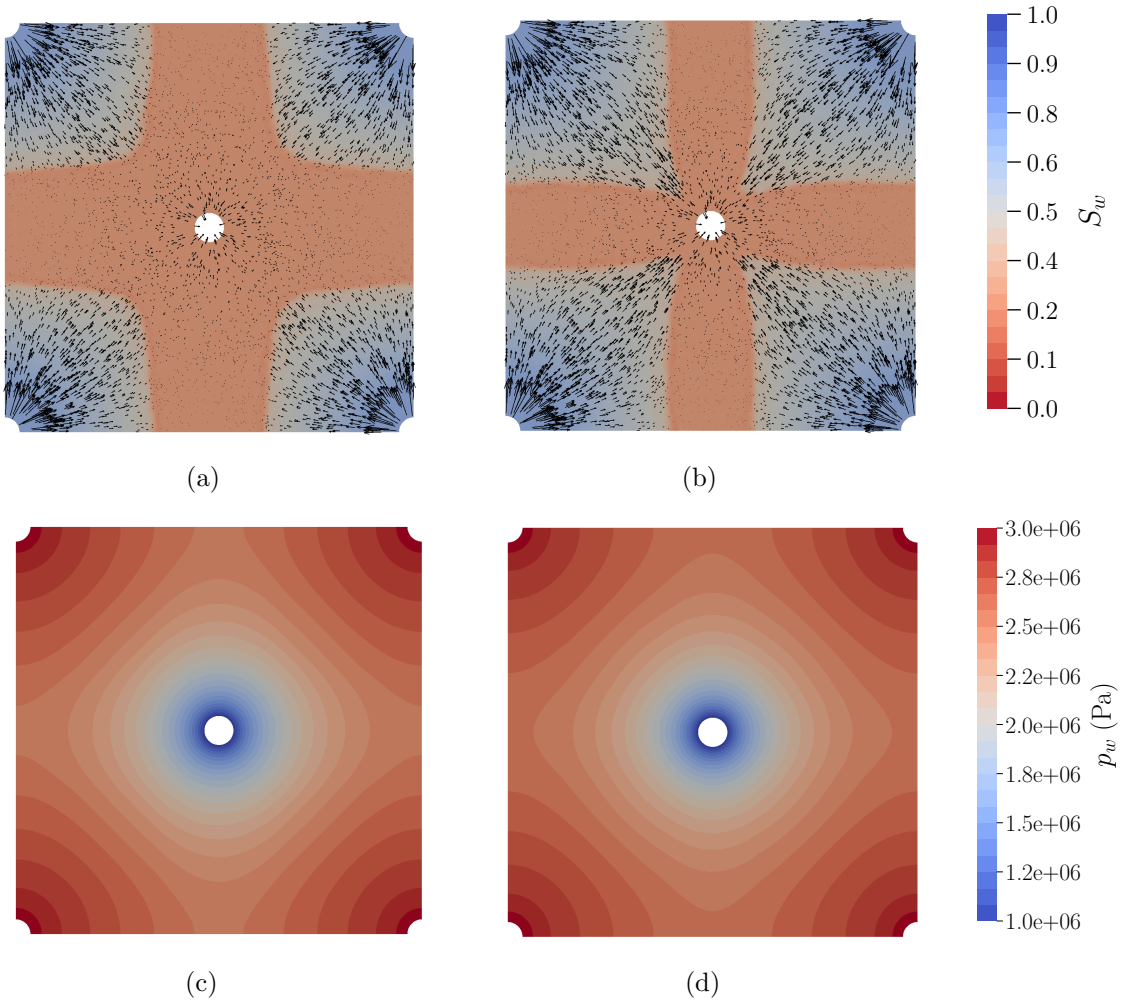


Figure 4.3: Water saturation approximation at time: a) 51 hours, and b) 65 hours. Water pressure approximation at time: c) 51 hours, and d) 65 hours.

and lower values at the extractor well.

4.7.3 Waterflooding through heterogeneous material

In this example, we consider a domain composed of two different permeability regions, Figure 4.2(b). The simulation domain, Ω, is a square of 100 m $\times$ 100 m, and its boundary is $\partial\Omega = \Gamma_{in} \bigcup \Gamma_{out} \bigcup \Gamma_{nf}$, see Figure 4.2(b). We prescribe the boundary conditions detailed in Equation (4.22) with $g_{Ds}^{in} = 0.22$.

The soil permeability is $\mathbf{K}_A = 10^{-13}\mathbf{I}$ m^2 and $\mathbf{K}_B = 10^{-12}\mathbf{I}$ m^2, the porosity is $\phi = 0.2$ and the viscosity for the water and oil phases are $\mu_w = 0.001$ Pa $\cdot$ s and

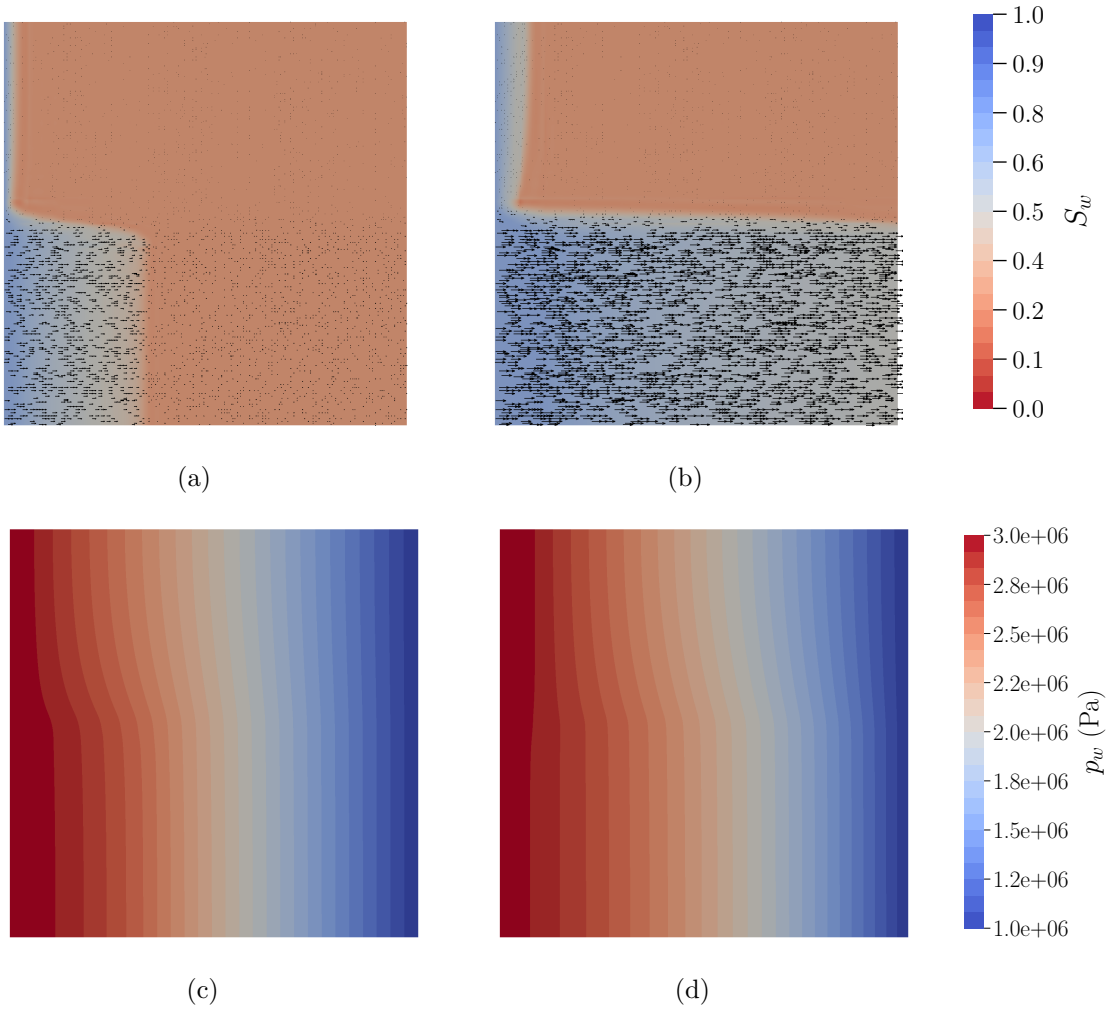


Figure 4.4: Water saturation approximation at time: a) 30 days, and b) 100 days. Water pressure approximation at time: c) 30 days, and d) 100 days.

$\mu_o = 0.012$ Pa $\cdot$ s, respectively.

We discretize the domain using 256 quadrilateral elements of polynomial degree four (4225 nodes), see Figure 4.2(b). The number of unknowns involved in the linear systems that have to be solved at each iteration of Equations (4.17) and (4.18) is 21920. Nevertheless, applying a static condensation procedure in the HDG formulation it is reduced to 2720 unknowns for each one. The time step for the DIRK3-s3 scheme is $\Delta t = 6$ h.

Figure 4.4 shows the computed water saturation and water pressure at time $t = 30$ days and $t = 100$ days. In Figures 4.4(a) and 4.4(b), we show the water saturation

and the water velocity vectors. At the beginning, the oil saturates homogeneously the porous media. Then, water is injected along the inflow boundary and moves the oil towards the outflow boundary. At both time steps the water is moving faster within the bottom region, where the permeability is higher, than in the upper region. Figures 4.4(c) and 4.4(d), show the water pressure field at the same time steps. Highest water pressure values are on the inflow boundary and the lowest on the outflow boundary. Moreover, when the injected water reaches to the outflow boundary, the water pressure in the entrance is equalized in both regions, and the water starts to saturate the upper region, as can be observed in Figures 4.4(b) and 4.4(d).

4.8 Concluding remarks

In this chapter, we have developed a memory-efficient high-order HDG formulation to solve the incompressible and immiscible two-phase flow in porous media problem. To this end, we have rewritten the initial second-order PDEs as a set of first-order PDEs, and we have deduced the weak form of the problem. We have computed the stabilization parameters in terms of the oil and total phase mobilities and the maximum eigenvalue of the permeability matrix. We have provided a framework to use high-order time discretization with a fix-point iteration method. In this framework, we compute both the pressure and saturation implicitly. The proposed non-linear solver reduces the memory footprint since it decouples the saturation and the pressure equations. Since we are using high-order DIRK schemes, we alternatively solve for the saturation and pressure unknowns stage-by-stage, which further reduces the memory consumption. The proposed non-linear solver has the same memory footprint as the IMPES method. However, the proposed solver solves implicitly both equations, while the IMPES solves the saturation explicitly. Moreover, our approach is more accurate than the classical IMPES since it is in a fix-point framework, in which we alternatively solve for both equations until we achieve convergence, while the classical IMPES does not have any iterative process.

We present several different examples to illustrate the capabilities of the proposed formulation and methodology. We have shown that the proposed formulation is high-accurate by validating the convergence rates for the scalar variables, their fluxes and their corresponding post-processed variables. Moreover, our formulation has been applied to solve the waterflooding technique for oil recovery simulation in two different

cases. Specifically, these two examples deals with several wells, heterogeneous material properties and unstructured meshes. We have seen that the number of iterations of the fix-point procedure depends on the complexity of the domain, the size of the element, and the time step. For the examples presented in this paper besides the convergence study, the number of iterations of the fix-point solver is around thirty.

Chapter 5

High-order HDG formulation with fully implicit temporal schemes for the simulation of two-phase flow through porous media

5.1 Introduction

In Chapter 4, we have coupled a high-order HDG formulation with high-order DIRK schemes. If high temporal accuracy is required, we can reduce the time step or increase the integration order. One the one hand, small time steps increase the computational cost because a non-linear problem has to be solve at each time step. On the other hand, it is known that stable high-order DIRK schemes need a high number of stages, especially above fourth-order. Thus, the computational cost of the simulation can be severely hampered because a non-linear problem has to be solved at each stage of each time step.

For that reason, in this chapter, we extend the proposed formulation for the immiscible and incompressible two-phase flow problem to perform the time integration with fully implicit RK schemes. These schemes are unconditionally stable and achieve high-order accuracy with few stages (Pazner and Persson, 2017). However, coupling high-order spatial discretization with fully-implicit high-order temporal schemes leads

to a denser and larger Jacobian matrix, which severely increases the memory requirements of the formulation (Pazner and Persson, 2017).

For non-smooth solutions, spurious oscillations may appear near the shocks or discontinuities (Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). These spurious oscillations compromise the accuracy of the numerical solution. Moreover, if high-order temporal schemes are used, these oscillations are not dissipated because of the low dissipation error of these schemes. In addition, the spurious oscillations may lead to non-physical values, like negative saturation, in which the physical model cannot be evaluated and the simulation has to be stopped.

To overcome these issues, the contributions of this chapter are the following:

1. **To propose a memory-efficient high-order HDG formulation for incompressible and immiscible two-phase flow through porous media coupled with high-order fully Runge-Kutta schemes.** To exploit the advantages of the high-order HDG formulation, we propose to perform a high-order fully implicit RK method to control the temporal error. Specifically, we use the Radau IIA and Gauss-Legendre schemes, which are unconditionally stable, achieve high-order temporal accuracy with few stages, and do not suffer order-reduction for two-phase flow through porous media problems. To reduce the memory consumption of coupling high-order spatial and temporal discretizations, we rewrite the non-linear system. We propose the temporal integration scheme in terms of the oil saturation instead of its temporal derivative. In this way, we obtain a better sparsity pattern in the Jacobian matrix and less coupling between the stages (Pazner and Persson, 2017). To further reduce the memory consumption, we adapt the fix-point iterative method proposed in Chapter 4 to deal with fully implicit RK schemes. This non-linear solver decouples the saturation and pressure systems. Therefore, we do not need to solve a non-linear system composed of both unknowns. We first solve for the saturation at all the stages in a single non-linear system using the Newton-Raphson method. Afterwards, we solve the pressure unknowns sequentially at each Runge-Kutta stage, since the pressure system does not couple the unknowns at different stages.

2. **To develop a robust and efficient shock capturing methodology.** To reduce the spurious oscillations at the vicinity of the sharp fronts, we introduce local artificial viscosity in the saturation equation (Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). The oscillations are detected using

the shock sensor introduced in Persson and Peraire (2006), but computed from the saturation solution and the post-processed saturation of HDG. Therefore, the proposed shock sensor is computationally efficient since the post-processed saturation is computed in an element-wise manner. Note that, in Persson and Peraire (2006) the sensor is computed by a comparison of two solutions of the problem computed with a different polynomial degree, P and $P-1$. Using the proposed shock sensor we do not need to solve again the problem with a different polynomial degree since we are using the HDG post-processed saturation. Furthermore, we compute the saturation and the post-processed saturation at the RK stages, and therefore, we obtain the shock sensor at the RK stages. Thus, this methodology allows us introducing a different amount of artificial viscosity at each RK stage and better tracking the sharp fronts as they evolve.

This chapter extends the proposed formulation in Chapter 4. Therefore, the numerical model and the corresponding HDG spatial discretization are detailed in Sections 4.2 and 4.3, respectively. The outline of this chapter is as follows. In Section 5.2, we couple the high-order HDG formulation with high-order fully implicit temporal discretization schemes. In Section 5.3, we detail the proposed non-linear solver. In Section 5.4, we state the local post-processing procedure. In Section 5.5, we specify the proposed shock capturing method. In Section 5.6, we present several examples to asses the capabilities of the proposed formulation. Finally, in Section 5.7, we summarize the main contributions of this chapter.

5.2 Time discretization

In Chapter 4, we have proposed to couple the high-order HDG with DIRK schemes for the two-phase flow through porous media problem. To this end, we have introduced the governing equations and the corresponding weak forms for this problem, see Sections 4.2 and 4.3, respectively. Moreover, in Chapter 4, we have obtained the corresponding first-order DAE by introducing the approximations for the oil saturation and water pressure unknowns into the corresponding weak forms, see Equations (4.8) and (4.9). Afterwards, in Section 4.4, we have detailed the temporal discretization using DIRK schemes.

In this section, we couple the proposed high-order HDG formulation in Chapter 4 with fully implicit RK schemes. To this end, we rewrite the obtained first-order DAE

in a compact form as in Chapter 4:

$$\mathbf{R}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) = \begin{bmatrix} \mathbf{R}_{S_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\hat{S}_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{p_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\mathbf{q}_p}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\hat{p}_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \end{bmatrix} = \mathbf{0}, \quad (5.1)$$

where $\mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p$, and $\hat{\mathbf{p}}_w$ are the time dependent nodal values of the unknowns.

To solve the DAE in Equation (5.1), we use a fully implicit RK method. As in Chapter 4, we denote by $(\cdot)^n$ the value of any variable at time t^n and by $(\cdot)^{n,i}$ the value of any variable at time $t^{n,i} = t^n + c^i \Delta t$, being n the time step and i the RK stage. Accordingly, we compute the oil saturation at time $t^{n+1} = t^n + \Delta t$ as

$$\mathbf{S}_o^{n+1} = \mathbf{S}_o^n + \Delta t \sum_{i=1}^{s} b^i \dot{\mathbf{S}}_o^{n,i}, \quad (5.2)$$

where $\dot{\mathbf{S}}_o^{n,i}$ is the approximation of $\dot{\mathbf{S}}_o$ at time $t^{n,i}$, and s is the total number of stages. We compute $\dot{\mathbf{S}}_o^{n,i}$ as the solution of the non-linear algebraic equations:

$$\mathbf{R}\left(t^{n,i}, \mathbf{S}_o^{n,i}, \dot{\mathbf{S}}_o^{n,i}, \mathbf{q}_s^{n,i}, \hat{\mathbf{S}}_o^{n,i}, \mathbf{p}_w^{n,i}, \mathbf{q}_p^{n,i}, \hat{\mathbf{p}}_w^{n,i}\right) = \mathbf{0}, \quad (5.3)$$

for $i = 1, \ldots, s$, where the oil saturation at each stage of the RK scheme, $\mathbf{S}_o^{n,i}$, is approximated using $\dot{\mathbf{S}}_o^{n,j}$ as

$$\mathbf{S}_o^{n,i} = \mathbf{S}_o^n + \Delta t \sum_{j=1}^{s} a^{ij} \dot{\mathbf{S}}_o^{n,j}. \quad (5.4)$$

The parameters a^{ij}, b^i, c^i define the RK method, and are given by the Butcher's tables (Butcher, 1964a,b, 2016; Montlaur et al., 2012; Kennedy and Carpenter, 2016; Pazner and Persson, 2017):

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b} \end{array} \equiv \begin{array}{c|ccc} c^1 & a^{11} & a^{12} & \ldots & a^{1s} \\ c^2 & a^{21} & a^{22} & \ldots & a^{2s} \\ \vdots & \vdots & & \ddots & \vdots \\ c^s & a^{s1} & & \ldots & a^{ss} \\ \hline & b^1 & b^2 & \ldots & b^s \end{array} .$$

Instead of solving Equation (5.3) for $\dot{\mathbf{S}}_o^{n,i}$, we propose to solve for $\mathbf{S}_o^{n,i}$. In this way, we obtain a better sparsity pattern in the Jacobian matrix and less coupling between stages (Pazner and Persson, 2017). To this end, we first rewrite Equation (5.4) as

$$\dot{\mathbf{S}}_o^{n,i} = \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\left(\mathbf{S}_o^{n,j} - \mathbf{S}_o^{n}\right), \tag{5.5}$$

where $\tilde{a}^{ij} = (\mathbf{A}^{-1})^{ij}$. Afterwards, we rewrite Equation (5.5) as

$$\dot{\mathbf{S}}_o^{n,i} = \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\mathbf{S}_o^{n,j} - \frac{\tilde{c}^{i}}{\Delta t}\mathbf{S}_o^{n}, \tag{5.6}$$

where $\tilde{c}^i = \sum_{j=1}^{s} \tilde{a}^{ij}$. Thus, inserting Equation (5.6) into Equation (5.3) we obtain the following non-linear algebraic equation:

$$\mathbf{R}\left(t^{n,i}, \mathbf{S}_o^{n,i}, \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\mathbf{S}_o^{n,j} - \frac{\tilde{c}^{i}}{\Delta t}\mathbf{S}_o^{n}, \mathbf{q}_s^{n,i}, \hat{\mathbf{S}}_o^{n,i}, \mathbf{p}_w^{n,i}, \mathbf{q}_p^{n,i}, \hat{\mathbf{p}}_w^{n,i}\right) = \mathbf{0}, \tag{5.7}$$

for $i = 1, \ldots, s$. Once the oil saturation is computed at all stages, we compute the oil saturation at next time step by inserting Equation (5.5) into Equation (5.2):

$$\mathbf{S}_o^{n+1} = \tilde{d}\mathbf{S}_o^{n} + \sum_{j=1}^{s} \tilde{b}^{j}\mathbf{S}_o^{n,j},$$

where $\tilde{d} = 1 - \sum_{j=1}^{s} \tilde{b}^{j}$ and $\tilde{\mathbf{b}} = \mathbf{b}\mathbf{A}^{-1}$.

5.3 Non-linear solver

To solve Equation (5.7) we use a fix-point iterative method. The main idea is to iteratively solve the saturation and the pressure unknowns until convergence is achieved, see Algorithm 5.1. Let l be the l-th iteration of the fix-point iterative method. Thus, we first solve Equation (5.7) for the oil saturation unknowns $(\mathbf{S}_o^{n,i,l+1}, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1})$ given the pressure unknowns $(\mathbf{p}_w^{n,i,l}, \mathbf{q}_p^{n,i,l}, \hat{\mathbf{p}}_w^{n,i,l})$, Line 6 of Algorithm 5.1. That is,

$$\mathbf{R}\Bigg(\underbrace{t^{n,i}}_{\text{Data}}, \underbrace{\mathbf{S}_o^{n,i,l+1}, \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\mathbf{S}_o^{n,j,l+1}}_{\text{Unknowns}} - \underbrace{\frac{\tilde{c}^{i}}{\Delta t}\mathbf{S}_o^{n}}_{\text{Data}}, \underbrace{\mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{\text{Unknowns}}, \underbrace{\mathbf{p}_w^{n,i,l}, \mathbf{q}_p^{n,i,l}, \hat{\mathbf{p}}_w^{n,i,l}}_{\text{Data}}\Bigg) = \mathbf{0}, \tag{5.8}$$

for $i = 1, \ldots, s$.

Once the saturation unknowns at all stages are computed, we solve Equation (5.7) for the water pressure unknowns $(\mathbf{p}_w^{n,i,l+1}, \mathbf{q}_p^{n,i,l+1}, \hat{\mathbf{p}}_w^{n,i,l+1})$, Line 7 of Algorithm 5.1, by imposing:

$$\mathbf{R}\Bigg(\underbrace{t^{n,i}, \mathbf{S}_o^{n,i,l+1}, \frac{1}{\Delta t}\sum_{j=1}^{s}\tilde{a}^{ij}\mathbf{S}_o^{n,j,l+1} - \frac{\tilde{c}^i}{\Delta t}\mathbf{S}_o^n, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{\text{Data}}, \underbrace{\mathbf{p}_w^{n,i,l+1}, \mathbf{q}_p^{n,i,l+1}, \hat{\mathbf{p}}_w^{n,i,l+1}}_{\text{Unknowns}}\Bigg) = \mathbf{0}, \tag{5.9}$$

for $i = 1, \ldots, s$.

We repeat this procedure until convergence is achieved for all Runge-Kutta stages, $i = 1, \ldots, s$, Line 9 of Algorithm 5.1. We define the stopping criteria of the non-linear solver using appropriate tolerances as

$$\begin{aligned}
\frac{\|S_{o_h}^{n,i,l} - S_{o_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|S_{o_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{S_o}, & \frac{\|p_{w_h}^{n,i,l} - p_{w_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|p_{w_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{p_w},\\
\frac{\|\mathbf{q}_{s_h}^{n,i,l} - \mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|\mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{\mathbf{q}_s}, & \frac{\|\mathbf{q}_{p_h}^{n,i,l} - \mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}}{\|\mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\mathrm{T}_h)}} &< \varepsilon_{\mathbf{q}_p},\\
\frac{\|\hat{S}_{o_h}^{n,i,l} - \hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}}{\|\hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}} &< \varepsilon_{\hat{S}_o}, & \frac{\|\hat{p}_{w_h}^{n,i,l} - \hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}}{\|\hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Sigma_h)}} &< \varepsilon_{\hat{p}_w},\\
\|\mathbf{R}_{S_o}\|_2 &< \varepsilon_{\mathbf{R}_{S_o}}, & \|\mathbf{R}_{p_w}\|_2 &< \varepsilon_{\mathbf{R}_{p_w}},\\
\|\mathbf{R}_{\mathbf{q}_s}\|_2 &< \varepsilon_{\mathbf{R}_{\mathbf{q}_s}}, & \|\mathbf{R}_{\mathbf{q}_p}\|_2 &< \varepsilon_{\mathbf{R}_{\mathbf{q}_p}},\\
\|\mathbf{R}_{\hat{S}_o}\|_2 &< \varepsilon_{\mathbf{R}_{\hat{S}_o}}, & \|\mathbf{R}_{\hat{p}_w}\|_2 &< \varepsilon_{\mathbf{R}_{\hat{p}_w}},
\end{aligned} \tag{5.10}$$

where $\|\cdot\|_{L^2(\mathrm{T}_h)}$ is the norm of the $L^2(\mathrm{T}_h)$ space of functions, $\|\cdot\|_{L^2(\Sigma_h)}$ is the norm of the $L^2(\Sigma_h)$ space of functions, and $\|\cdot\|_2$ is the Euclidean norm of vectors.

We highlight that for each iteration of the fix-point method, we solve a non-linear system for the saturation, Equation (5.8), since the λ_o, λ_t and p_c' depend on $\mathbf{S}_o^{n,i,l+1}$. We solve the saturation system using the Newton-Raphson method. Since all the stages are coupled, this non-linear system has $s \times (n_{S_o} + n_{\mathbf{q}_s} + n_{\hat{S}_o})$ unknowns, being s the number of RK stages, and n_S, $n_{\mathbf{q}_s}$ and $n_{\hat{S}_o}$ the number of unknowns for the oil saturation, the saturation flux and the oil saturation traces, respectively.

Once we obtain an approximation for the oil saturation unknowns at all the stages, we compute the water pressure unknowns by solving s uncoupled linear systems of size $n_{p_w} + n_{\mathbf{q}_p} + n_{\hat{p}_w}$, where n_{p_w}, $n_{\mathbf{q}_p}$ and $n_{\hat{p}_w}$ are the number of unknowns for the water pressure, the pressure flux and the water pressure traces, respectively.

Algorithm 5.1 Fix-point iteration method for two-phase flow.

1: **Input**: $\mathbf{S}_o^n$
2: $\mathbf{S}_o^{n,0} = \mathbf{S}_o^{n-1}$, $\mathbf{p}_w^{n,0} = \mathbf{p}_w^{n-1}$.
3: l = 0
4: convergence = False
5: **While** (not convergence):
6: **Compute**: $\mathbf{S}_o^{n,i,l+1}$, $\mathbf{q}_s^{n,i,l+1}$ and $\hat{\mathbf{S}}_o^{n,i,l+1}$ from $\mathbf{S}_o^n$, $\mathbf{p}_w^{n,l}$, $\mathbf{q}_p^{n,i,l}$ and $\hat{\mathbf{p}}_w^{n,i,l}$ using Eq.(5.8).
7: **Compute**: $\mathbf{p}_w^{n,i,l+1}$, $\mathbf{q}_p^{n,i,l+1}$ and $\hat{\mathbf{p}}_w^{n,i,l+1}$ from $\mathbf{S}_o^{n,l+1}$, $\mathbf{q}_s^{n,i,l+1}$ and $\hat{\mathbf{S}}_o^{n,i,l+1}$ using Eq.(5.9).
8: $l = l + 1$
9: **check convergence** using Eq.(5.10).
10: $\mathbf{S}_o^{n,i} = \mathbf{S}_o^{n,i,l}$, $\mathbf{q}_s^{n,i} = \mathbf{q}_s^{n,i,l}$, $\hat{\mathbf{S}}_o^{n,i} = \hat{\mathbf{S}}_o^{n,i,l}$, $\mathbf{p}_w^{n,i} = \mathbf{p}_w^{n,i,l}$, $\mathbf{q}_p^{n,i} = \mathbf{q}_p^{n,i,l}$, $\hat{\mathbf{p}}_w^{n,i} = \hat{\mathbf{p}}_w^{n,i,l}$.
11: end

Furthermore, the global linear system for the saturation unknowns and the linear systems for the pressure unknowns are hybridized in terms of $\hat{\mathbf{S}}_o^{n,i,l+1}$ and $\hat{\mathbf{p}}_w^{n,i,l+1}$ respectively (Nguyen et al., 2009a,b; Kirby et al., 2012; Sevilla and Huerta, 2016). The other unknowns $\mathbf{S}_o^{n,i,l+1}$, $\mathbf{q}_s^{n,i,l+1}$ $\mathbf{p}_w^{n,i,l+1}$ and $\mathbf{q}_p^{n,i,l+1}$ are recovered using an element-wise process. The hybridization procedure is applied in the saturation and pressure systems similarly as in Section 3.6 of Chapter 3.

This non-linear solver is memory-efficient since the saturation and the pressure equations are decoupled. Therefore, there is no need to solve for both unknowns at the same time. Furthermore, the pressure equation is solved sequentially stage by stage, because the pressure is not coupled at different stages.

5.3.1 Saturation solver

We use the Newton-Raphson method to solve the saturation system, Equation (5.8). From now on, to ease the notation, we drop the super-index l corresponding to the fix-point iteration. We concatenate all the oil saturation unknowns of all stages in a vector of unknowns $\mathbf{u}^n$ as

$$\mathbf{u}^n = \begin{bmatrix} \mathbf{u}^{n,1} \\ \vdots \\ \mathbf{u}^{n,s} \end{bmatrix}, \quad \text{where} \qquad \mathbf{u}^{n,i} = \begin{bmatrix} \mathbf{S}_o^{n,i} \\ \mathbf{q}_s^{n,i} \\ \hat{\mathbf{S}}_o^{n,i} \end{bmatrix}, \quad \text{for} \;\; i = 1 \dots s.$$

Thus, the non-linear residual of the saturation is

$$\mathbf{F}\left(\mathbf{u}^n\right) = \begin{bmatrix} \mathbf{R}_S\left(t^{n,1}, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t^{n,1}, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \mathbf{R}_{\hat{S}}\left(t^{n,1}, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \vdots \\ \mathbf{R}_S\left(t^{n,s}, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t^{n,s}, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \\ \mathbf{R}_{\hat{S}}\left(t^{n,s}, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \end{bmatrix}.$$

The Newton-Raphson method involves successive approximations of the solution $\mathbf{u}^n$. The $k+1$ approximation is obtained as

$$\mathbf{u}^{n,k+1} = \mathbf{u}^{n,k} + \delta\mathbf{u}^{n,k},$$

where the supra-index k denotes the Newton-Raphson iteration and $\delta\mathbf{u}^{n,k}$ is the solution of the linear system

$$\mathbf{J}\left(\mathbf{u}^{n,k}\right)\delta\mathbf{u}^{n,k} = -\mathbf{F}\left(\mathbf{u}^{n,k}\right),$$

being $\mathbf{J}\left(\mathbf{u}^{n,k}\right)$ the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{u}^{n,k}$. The Jacobian matrix is a block sparse matrix, in which each block corresponds to a RK stage. We decompose $\mathbf{J}$ as the summation of two matrices:

$$\mathbf{J} = \begin{bmatrix} \mathbf{M}^{11} & \mathbf{M}^{12} & \dots & \mathbf{M}^{1s} \\ \mathbf{M}^{21} & \mathbf{M}^{22} & \dots & \mathbf{M}^{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}^{s1} & \mathbf{M}^{s2} & \dots & \mathbf{M}^{ss} \end{bmatrix} + \begin{bmatrix} \mathbf{J}^{11} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{22} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}^{ss} \end{bmatrix}. \tag{5.11}$$

The first matrix in Equation (5.11) comes from the term containing the time derivative of the oil saturation. Since it couples all the unknowns $\mathbf{S}_o^{n,i}$ for all the stages, it has contribution in all the blocks. Each block $\mathbf{M}^{ij}$ for $i,j = 1 \dots s$ is also a block matrix defined as

$$\mathbf{M}^{ij} = \begin{bmatrix} \mathbf{M}\dfrac{\tilde{a}^{ij}}{\Delta t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $[\mathbf{M}]_{k,l} = (\phi N_k, N_l)$. The matrices $\mathbf{M}$ are constant in all the stages, since the porosity, ϕ, does not depend on time. Therefore, the matrices $\mathbf{M}$ are computed only once and are reused in all the stages.

The second matrix of Equation (5.11) is a block-diagonal matrix. This matrix does not couple the unknowns of different stages. The block diagonal matrices are not constant in the stages since they depend on the saturation. Nevertheless, there are some elemental contributions that are constant and can be reused in the different stages. The matrices $\mathbf{J}^{ii}$ for $i = 1 \dots s$ are equivalent to the matrices obtained when solving a two-phase flow stationary problem.

Note that it is not necessary to converge the Newton-Raphson method for the saturation unknowns because the pressure unknowns are not converged yet. Therefore, to reduce the computational cost, we perform only one iteration of the Newton-Raphson method.

5.3.2 Pressure solver

The pressure unknowns of the different stages are not coupled because the pressure equation does not contain a temporal part, see Equation (5.9). Thus, we solve s uncoupled linear systems and we reduce the memory footprint and the computational cost of solving the pressure equation. Moreover, since we are using the fix-point iterative method described in Algorithm 5.1, the pressure equation is linear because the unknowns related to the saturation are considered as parameters in the pressure equation.

5.4 Local post-processing

One of the main advantages of using the HDG formulation is that the scalar variables (water pressure and oil saturation), and their fluxes have a rate of convergence of $P + 1$ in the L^2-norm. Moreover, a local post-processing can be applied to obtain a new approximation for the saturation, $S^*_{o_h}$, and for the pressure, $p^*_{w_h}$, both in $\mathcal{V}_h^{P+1}$ with convergence rate of $P + 2$ in the L^2-norm (Nguyen et al., 2009a,b, 2011, 2013; Kirby et al., 2012; Roca et al., 2013; Giorgiani et al., 2013, 2014; Sevilla and Huerta, 2016; Paipuri et al., 2018).

We apply the local post-processing at the stages of the RK scheme. The local problems of the elements at each stage are independent and can be solved separately.

In our formulation, we have two local problems, one for the oil saturation, $S^*_{o_h}$, and other for the water pressure, $p^*_{w_h}$.

The first local problem consist of finding the post-processed saturation, $S^*_{o_h} \in \mathcal{V}_h^{P+1}$ on each element, e, and at all the stages of the RK scheme, such that:

$$\begin{aligned}(\mathbf{K}\lambda_o p_c' \nabla S^*_{o_h}, \nabla v)_e = & -(\mathbf{q}_{s_h}, \nabla v)_e \\ (S^*_{o_h}, 1)_e = & (S_{o_h}, 1)_e,\end{aligned} \tag{5.12}$$

for all $v \in \mathcal{V}_h^{P+1}$. Equation (5.12) is non-linear since λ_o and p_c' depend on $S^*_{o_h}$. To solve Equation (5.12), we apply the Newton-Raphson method. Once we find the post-processed saturation, $S^*_{o_h}$, we solve the second local problem to find the post-processed water pressure, $p^*_{w_h} \in \mathcal{V}_h^{P+1}$ on each element, e, and at all the stages of the RK scheme. Specifically, we solve

$$\begin{aligned}(\mathbf{K}\lambda_t \nabla p^*_{w_h}, \nabla v)_e = & -(\mathbf{q}_{p_h}, \nabla v)_e \\ (p^*_{w_h}, 1)_e = & (p_{w_h}, 1)_e,\end{aligned} \tag{5.13}$$

for all $v \in \mathcal{V}_h^{P+1}$. Note that this is a linear problem in each element and at each stage because the saturation at all stages is known.

5.5 Shock capturing

High-order methods, such HDG method, are high-accurate if the solution is smooth enough. Nevertheless, if there is a sharp front (or discontinuity) in the solution, oscillations may appear at the vicinity of the front (VonNeumann and Richtmyer, 1950; Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). These oscillations may introduce spurious artifacts that hamper the accuracy of the obtained solution and the robustness of the solver.

To reduce these spurious oscillations we introduce artificial viscosity (VonNeumann and Richtmyer, 1950; Persson and Peraire, 2006; Huerta et al., 2012; Casoni et al., 2013). We identify the elements containing the sharp front using the shock sensor introduced in Persson and Peraire (2006). The main difference with Persson and Peraire (2006) is that we compute the shock sensor from the saturation solution and the post-processed saturation of HDG, $S^*_{o_h}$, obtained in Equation (5.12). Therefore, the shock sensor is computed in an efficient manner as

$$S_e = \frac{(S_{o_h} - S^*_{o_h}, S_{o_h} - S^*_{o_h})_e}{(S^*_{o_h}, S^*_{o_h})_e}. \tag{5.14}$$

The main advantage of using the post-processed saturation $S^*_{o_h}$ to compute the shock sensor is the reduction of the computational cost, since $S^*_{o_h}$ is obtained with an element-wise post-processing. Thus, we do not need to solve again the problem with a different polynomial degree. We compute the saturation and the post-processed saturation at the RK stages, and therefore, we obtain the shock sensor at the RK stages. Since the shock sensor is different at each RK stage, it allows tracking the sharp fronts at the different RK stages.

According to Persson and Peraire (2006), we define the artificial viscosity factor, ε, which is related to the resolution of the spatial discretization, as

$$\varepsilon = \begin{cases} 0 & \text{if } s_e < s_0 - \kappa, \\ \dfrac{\epsilon_0}{2}\left(1 + \sin\left(\dfrac{\pi(s_e - s_0)}{2\kappa}\right)\right) & \text{if } s_0 - \kappa \leq s_e \leq s_o + \kappa, \\ \epsilon_0 & \text{if } s_e > s_0 + \kappa, \end{cases} \tag{5.15}$$

where $s_e = \log_{10} S_e$, $\epsilon_0 \simeq \dfrac{h_e}{P}$, and s_o and κ are chosen empirically. Note that, as the polynomial degree increases, the artificial viscosity factor, ε, decreases, and as the element size increases the artificial viscosity factor, ε, increases. Note that, the artificial viscosity is also different at each RK stage, because the sharp fronts position evolves in time.

We add an artificial viscosity term to reduce the oscillations of those elements detected by the sensor. In this work, we only introduce the artificial viscosity term in the saturation equation because the pressure solution is smooth. Thus, we rewrite Equation 4.2 as

$$\phi\frac{\partial S_o}{\partial t} + \nabla\cdot\left(-\lambda_o\mathbf{K}p_c'\nabla S_o - \varepsilon\lambda_o\mathbf{K}p_c'\nabla S_o - \frac{\lambda_o}{\lambda_t}\lambda_t\mathbf{K}\nabla p_w\right) = f_o(\mathbf{x},t) \qquad \forall(\mathbf{x},t)\in(\Omega,T).$$

In terms of fist-order PDEs the system to be solved for the oil saturation is

$$\begin{cases} \phi\dfrac{\partial S_o}{\partial t} + \nabla\cdot\left(\mathbf{q}_{s,\varepsilon} + \dfrac{\lambda_o}{\lambda_t}\mathbf{q}_p\right) = f_o(\mathbf{x},t) & \forall(\mathbf{x},t)\in(\Omega,T), \\ \mathbf{q}_{s,\varepsilon} + \lambda_o\mathbf{K}_\varepsilon p_c'\nabla S_o = \mathbf{0} & \forall(\mathbf{x},t)\in(\Omega,T), \\ S_o^{\Gamma_{in}}(\mathbf{x},t) = g^{in}_{Ds}(\mathbf{x},t) & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ \left(\dfrac{\lambda_o\lambda_w}{\lambda_t}\mathbf{K}\nabla p_c\right)\cdot\mathbf{n} = g^{out}_{Ns}(\mathbf{x},t) & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_o\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T), \\ S_o(\cdot,0) = S_o^0(\mathbf{x}) & \forall\mathbf{x}\in\Omega, \end{cases} \tag{5.16}$$

and for the water pressure is

$$\begin{cases} \nabla \cdot \left(\mathbf{q}_p + \dfrac{\mathbf{q}_{s,\varepsilon}}{1+\varepsilon} \right) = f_o(\mathbf{x},t) + f_w(\mathbf{x},t) & \forall (\mathbf{x},t) \in (\Omega, T), \\ \mathbf{q}_p + \lambda_t \mathbf{K} \nabla p_w = \mathbf{0} & \forall (\mathbf{x},t) \in (\Omega, T), \\ {p_w}^{\Gamma_{in}}(\mathbf{x},t) = g_{Dp}^{in}(\mathbf{x},t) & \forall (\mathbf{x},t) \in (\Gamma_{in}, T), \\ {p_w}^{\Gamma_{out}}(\mathbf{x},t) = g_{Dp}^{out}(\mathbf{x},t) & \forall (\mathbf{x},t) \in (\Gamma_{out}, T), \\ \mathbf{v}_t \cdot \mathbf{n} = 0 & \forall (\mathbf{x},t) \in (\Gamma_{nf}, Ts). \end{cases} \tag{5.17}$$

For both systems, ε is the artificial viscosity factor, $\mathbf{q}_{s,\varepsilon} = (1+\varepsilon)\mathbf{q}_s$ is the new diffusive flux for the saturation equation, and $\mathbf{K}_\varepsilon = (1+\varepsilon)\mathbf{K}$ is the new permeability of the porous media for the saturation equation. Therefore, for $\varepsilon = 0$, we get $\mathbf{q}_{s,\varepsilon} = \mathbf{q}_s$ and $\mathbf{K}_\varepsilon = \mathbf{K}$, and we recover the original problem. Note that the water pressure system uses $\mathbf{q}_{s,\varepsilon}$, but we recover the same formulation as in Equation (4.7) of Chapter 4 since the pressure solution is smooth and does not require adding local artificial viscosity.

Adding artificial viscosity can be interpreted as increasing the permeability of the porous media in the detected elements. Therefore, we have to balance the amount of artificial viscosity to reduce the spurious oscillations without changing the underlying physics.

To introduce artificial viscosity, we need to compute the shock sensor that depends on the saturation and the post-processed saturation, see Equation (5.14). Therefore, we need to ensure that the temporal error is low enough to compute the post-processed saturation with enough accuracy. For this reason, we propose to use temporal integration schemes with, at least, the same convergence rate as the expected convergence rate of the post-processed saturation.

To compute the solution for a give time step, we perform a two-step method. First, we compute the solution without adding artificial viscosity. Second, we compute the amount of required artificial viscosity at each stage, and we compute the solution again. We highlight that the artificial viscosity is different at each RK stage since it depends on the shock sensor. Thus, the proposed methodology allows tracking the sharp front as it moves at different stages and introducing appropriate amount of artificial viscosity at each stage.

5.6 Examples

In this section, we present seven examples to assess the capabilities of the proposed formulation. Example 5.6.1 shows the convergence rates for the obtained solutions of the proposed formulation. Example 5.6.2 analyses the effect of introducing artificial viscosity. Examples 5.6.3 to 5.6.5 focus on the accuracy of high-order spatial and temporal discretization methods of the proposed formulation. Finally, Examples 5.6.6 and 5.6.7 present two cases of waterflooding simulation. The first one considers heterogeneous material properties and the second one analyses the nine-spot pattern.

For the Examples 5.6.2 to 5.6.7,we set the characteristic lengths in Equation (4.11) as $l_s = 10^{-7}$ and $l_p = 10^{-10}$. In addition, we set the following tolerances for the fix-point solver, see Equation (5.10):

$$\varepsilon_{S_o} = 10^{-8}, \varepsilon_{\mathbf{q}_s} = 10^{-6}, \quad \varepsilon_{\hat{S}_o} = 10^{-8}, \varepsilon_{\mathbf{R}_{S_o}} = 10^{-8}, \quad \varepsilon_{\mathbf{R}_{\mathbf{q}_s}} = 10^{-6}, \varepsilon_{\mathbf{R}_{\hat{S}_o}} = 10^{-8},$$
$$\varepsilon_{p_w} = 10^{-8}, \varepsilon_{\mathbf{q}_p} = 10^{-6}, \quad \varepsilon_{\hat{p}_w} = 10^{-8}, \varepsilon_{\mathbf{R}_{p_w}} = 10^{-8}, \quad \varepsilon_{\mathbf{R}_{\mathbf{q}_p}} = 10^{-6}, \varepsilon_{\mathbf{R}_{\hat{p}_w}} = 10^{-8}.$$

We also define $\varepsilon_{S^*_{o_h}} = 10^{-12}$ and $\varepsilon_{\mathbf{R}_{S^*_{o_h}}} = 10^{-6}$, as the relative and absolute tolerances of the Netwon-Raphson method to compute the local post-processing for the saturation, see Equation (5.12).

For all the examples, the Butcher tables of the used time integration schemes are detailed in Appendix B, and all the used high-order meshes are generated using the algorithms presented in Gargallo-Peiró et al. (2015, 2016); Ruiz-Gironés E.; Roca and Sarrate (2016), which are implemented in the EZ4U environment (Roca et al., 2010).

5.6.1 Convergence rate analysis

In this example, we analyze the convergence error in space and time of the numerical approximation obtained with the proposed method using an analytical solution. To this end, we define the space-time L^2-norm of a function as

$$\|u\left(\mathbf{x},t\right)\|^2_{L^2(\Omega,T)} = \int_0^{t_{end}} \int_\Omega \|u\left(\mathbf{x},t\right)\|^2 \mathrm{d}\Omega \; \mathrm{d}t.$$

We approximate the space-time integration as

$$\begin{aligned}\int_0^{t_{end}}\int_\Omega \|u\left(\mathbf{x},t\right)\|^2 \mathrm{d}\Omega\ \mathrm{d}t &= \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h}\int_e \|u\left(\mathbf{x},t\right)\|^2 \mathrm{d}\Omega\ \mathrm{d}t \\ &= \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h}\int_{e^M} \|u\left(\xi_g,t\right)\|^2 |J_g| \mathrm{d}\xi\ \mathrm{d}t \\ &\simeq \int_0^{t_{end}} \underbrace{\sum_{e\in \mathrm{T}_h}\sum_{g=1}^{N_g} \|u\left(\xi_g,t\right)\|^2 |J_g|\omega_g \mathrm{d}\xi}_{f(t)}\ \mathrm{d}t \\ &= \int_0^{t_{end}} f\left(t\right)\mathrm{d}t,\end{aligned} \tag{5.18}$$

where N_g is the number of integration points. To perfom the time integral we use the time integration schemes of the Runge-Kutta methods. That is, b^i are the temporal integration weights, and c^i are the temporal integration points for a time interval $(0,1)$. Thus,

$$\int_0^{t_{end}} f\left(t\right)\mathrm{d}t = \sum_n^{N_s}\int_{t^n}^{t^{n+1}} f\left(t\right)\mathrm{d}t \simeq \Delta t \sum_{n=1}^{N_s}\sum_{i=1}^{s} f\left(t^n + c^i\Delta t\right) b^i, \tag{5.19}$$

where N_s is the number of time steps. Afterwards, substituting Equation (5.19) into Equation (5.18) we obtain that the approximation of the space-time L^2-norm of a function is:

$$\begin{aligned}\|u\|^2_{L^2(\Omega,T)} &\simeq \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h}\sum_{g=1}^{N_g} \|u\left(\xi_g,t\right)\|^2 |J_g|\omega_g \mathrm{d}\xi\ \mathrm{d}t \\ &\simeq \sum_{n=1}^{N_s}\sum_{i=1}^{s}\sum_{e\in \mathrm{T}_h}\sum_{g=1}^{N_g} \|u\left(\xi_g,t^{n,i}\right)\|^2 |J_g|\omega_g b^i \Delta t,\end{aligned}$$

where $t^{n,i} = t^n + c^i\Delta t$.

We show numerical evidence of the convergence rates of the space-time error in L^2-norm for different polynomial degrees and time integration schemes for the oil saturation, S_{o_h}, the saturation flux, $\mathbf{q}_{s_h}$, the post-processed saturation, $S^*_{o_h}$, the water pressure, p_{w_h}, the pressure flux, $\mathbf{q}_{p_h}$, and the post-processed pressure, $p^*_{w_h}$. We define the error of each variable as

$$\begin{aligned} E_{S_o} &= \|S_{o_h} - S_o\|_{L^2(\Omega,T)}, \quad E_{\mathbf{q}_S} = \|\mathbf{q}_{S_h} - \mathbf{q}_S\|_{L^2(\Omega,T)}, \quad E_{S^*_o} = \|S^*_{o_h} - S^*_o\|_{L^2(\Omega,T)}, \\ E_{p_w} &= \|p_{w_h} - p_w\|_{L^2(\Omega,T)}, \quad E_{\mathbf{q}_p} = \|\mathbf{q}_{p_h} - \mathbf{q}_p\|_{L^2(\Omega,T)}, \quad E_{p^*_w} = \|p^*_{w_h} - p^*_w\|_{L^2(\Omega,T)}. \end{aligned}$$

To this end, we define the analytical saturation and pressure solutions

$$S_o = \sin(\pi x)\sin(\pi y)\sin(t), \qquad p_w = \cos(\pi x)\cos(\pi y)\cos(t), \tag{5.20}$$

where $(x, y) \in \Omega = (0, 2) \times (0, 2)$ and $t \in (0, 1)$.

We set the soil permeability as $\mathbf{K} = \mathbf{I}$ m^2, the porosity as $\phi = 0.1$, the oil viscosity as $\mu_o = 1$ Pa·s and the water viscosity as $\mu_w = 0.1$ Pa·s. Specifically for this example, we use the Brooks-Corey model, Equation (4.1), with $p_e = 0.5$ Pa and $\theta = 1$. We select $\tau_p = 10$ and $\tau_s = 10$, see Equation (4.11). We prescribe Dirichlet boundary conditions on the whole boundary and we set the source terms in order to obtain the analytical solutions defined in Equation (5.20). We do not introduce artificial viscosity since the analytical solution is smooth.

We generate a series of meshes composed of quadrilateral elements of polynomial degrees between one and five. All these meshes are combined with high-order GL schemes that converge with the same rate or higher than the post-processed variables. For each polynomial degree, we keep constant the ratio $h_e/\Delta t$.

We set to 10^{-10} all the tolerances of the fix-point solver, see Equation (5.10). We also set the $\varepsilon_{S^*_{o_h}} = 10^{-12}$ and $\varepsilon_{\mathbf{R}_{S^*_{o_h}}} = 10^{-10}$ tolerances for the Netwon-Raphson method to compute the local post-processing for the saturation, see Equation (5.12).

Figure 5.1 shows the convergence rates of the space-time error for the oil saturation and the pressure, their fluxes, and the post-processed solutions. We obtain the expected convergence rate of $P + 1$ in L^2-norm for the oil saturation, water pressure and for the fluxes $\mathbf{q}_s$ and $\mathbf{q}_p$. The local post-process, detailed in Equations (5.12) and (5.13), is applied to obtain a super convergence rate of $P + 2$ in L^2-norm of the post-processed saturation, $S^*_{o_h}$, and the post-processed pressure, $p^*_{w_h}$.

5.6.2 Artificial viscosity analysis

In this example, we analyze the behavior of the artificial viscosity term. We consider a rectangular domain, $\Omega = (0, 84) \times (0, 2h_e)$ meters, where h_e is the element size of the different spatial discretizations. Water is injected from the right side of the domain, Γ_{in}, and extracted for the left side, Γ_{out}. Furthermore, we assume that both fluids cannot cross the upper and lower boundaries, Γ_{nf}, see Figure 5.2. Therefore, water will mobilize the oil from left to right and the flux will be parallel to the no-flow

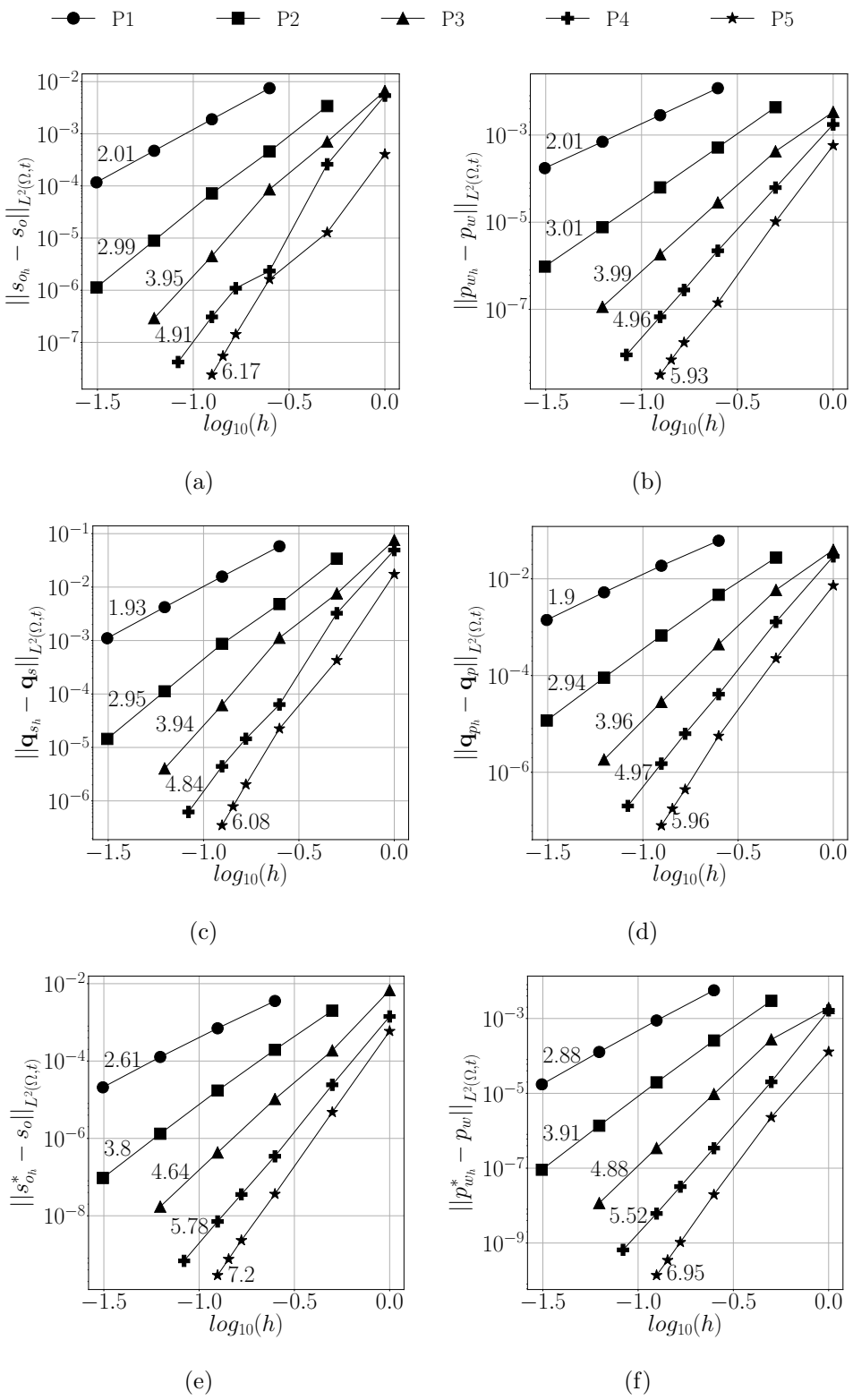


Figure 5.1: Convergence rates for: a) saturation, b) pressure, c) saturation flux, d) pressure flux; e) post-processed saturation, and f) post-processed pressure.

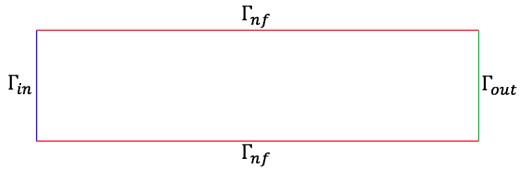


Figure 5.2: Rectangular domain and associated boundary conditions.

boundaries, Γ_{nf}. We define the following boundary conditions and source terms:

$$
\begin{array}{llllll}
p_w^{\Gamma_{in}} & = 3 \cdot 10^6 \text{ Pa}, & S_o^{\Gamma_{in}} & = 0.3, & \text{on } \Gamma_{in}, \\
p_w^{\Gamma_{out}} & = 10^6 \text{ Pa}, & \left(\dfrac{\lambda_o \lambda_w}{\lambda_t} \mathbf{K} \nabla p_c\right) \cdot \mathbf{n} & = 0, & \text{on } \Gamma_{out}, \\
\mathbf{v}_t \cdot \mathbf{n} & = 0, & \mathbf{v}_o \cdot \mathbf{n} & = 0, & \text{on } \Gamma_{n_f}, \\
f_w & = 0, & f_o & = 0, & \text{in } \Omega,
\end{array}
$$

and we set the initial oil saturation condition as $S_o^0 = 0.7$.

The parameters of the Brooks-Corey model are $p_e = 10^3$ Pa and $\theta = 2$, and the residuals saturation for the water and oil are $S_{rw} = 0$, $S_{ro} = 0$, respectively. The soil permeability is $\mathbf{K}_A = 10^{-12}\mathbf{I}$ m^2, the porosity is $\phi = 0.2$ and the viscosity for the water and oil phases are $\mu_w = 0.001$ Pa $\cdot$ s and $\mu_o = 0.012$ Pa $\cdot$ s, respectively.

We discretize the domain with quadrilateral elements of polynomial degree $P = 6$ with size $h_e = 10.5$ meters, and we use the GL8 scheme with a time step $\Delta t = 2$ days. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 0.0, 2.5, 5.0, 7.5$ and 10.0 in Equation (5.15).

Figure 5.3 shows the plot over the line $y = 0$ for the water saturation and water pressure for the selected ϵ_0 values at time 22 days. As expected, only the saturation approximation presents a sharp front, whereas the pressure approximation is smooth. As we increase the amount of artificial viscosity the oscillations and the discontinuities between elements are reduced. However the sharp front is dissipated and becomes less vertical, see Figures 5.3(a) and 5.3(b). This effect can be interpreted as a local increase of the intrinsic permeability in the saturation equation, $\mathbf{K}$, as it is shown in Equation (5.16) and (5.17). As a consequence of this permeability increment, we obtain a more diffused front. Note that for all the selected ϵ_0 values the water pressure does not have significant variations, see Figure 5.3(c).

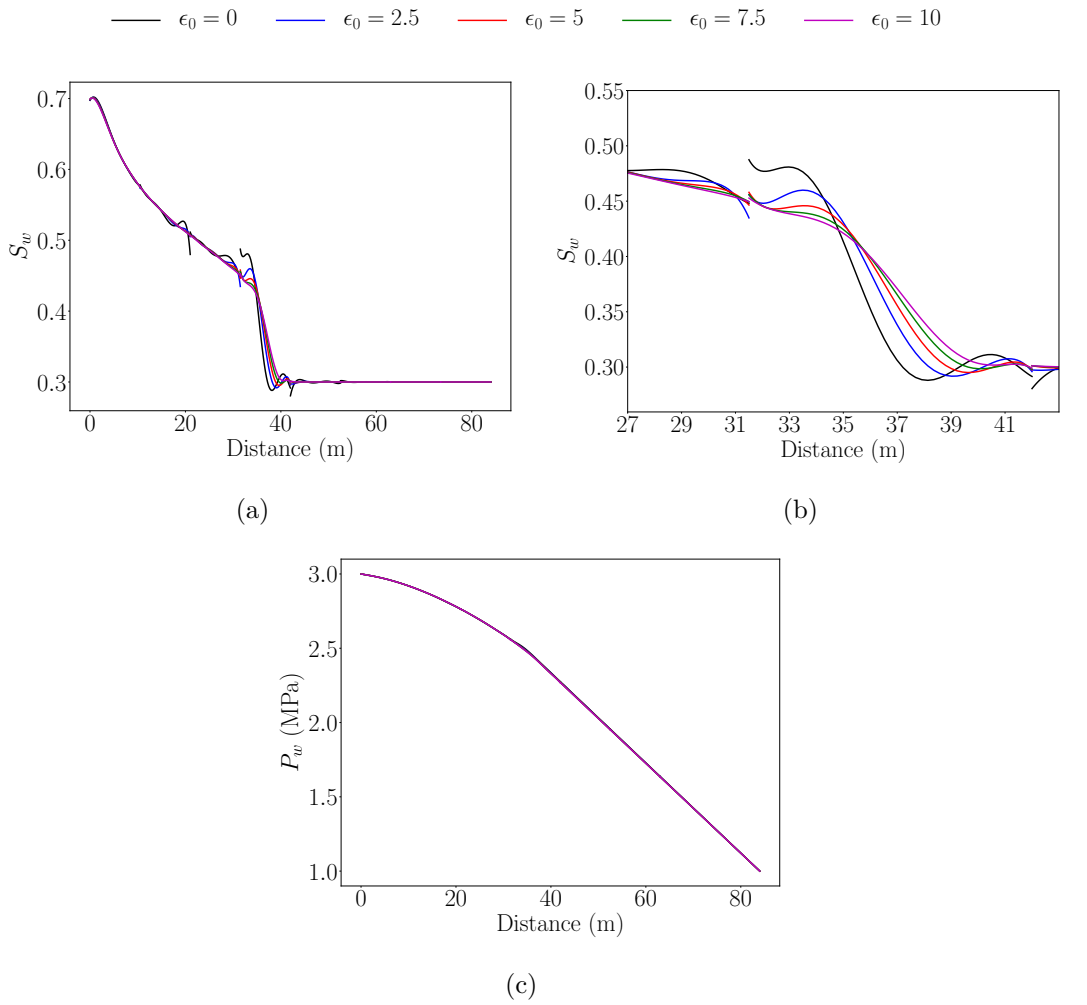
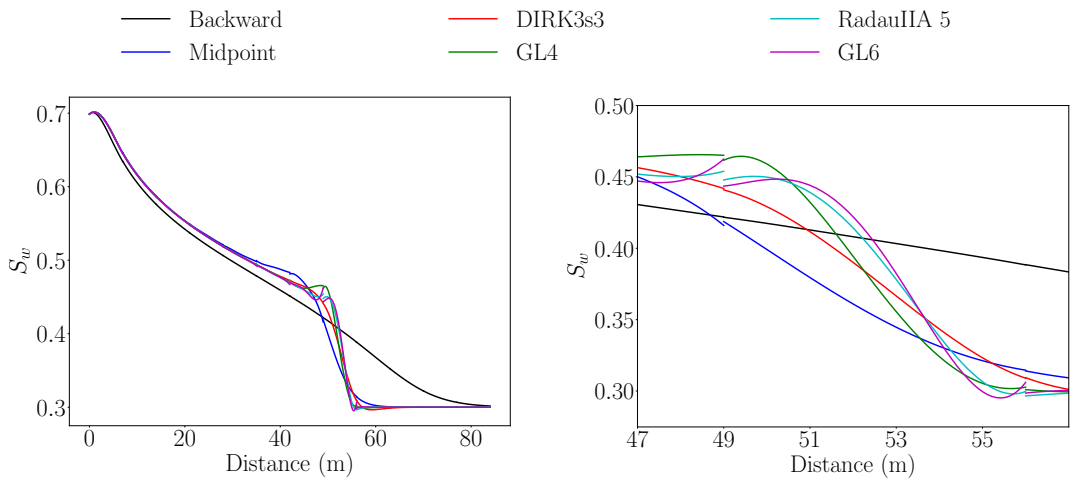


Figure 5.3: Plot over the line $y = 0$ at 22 days using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with $h_e = 10.5$ meters, and using GL8 scheme with $\Delta t = 2$ days for different values of ϵ_0: a) water saturation, b) zoom around the sharp front of the water saturation, and c) water pressure.

5.6.3 Time integration schemes analysis

In this example, we compare different time integration schemes for the same spatial discretization. We consider the same rectangular domain, boundary conditions and material parameters of Example 5.6.2. We use a mesh with quadrilateral elements of polynomial degree $P = 4$ with size $h_e = 7$ meters combined with the following time integration schemes: backward Euler, midpoint, DIRK3s3, GL4, Radau IIA 5 and GL6. These are time integration schemes of orders from one to six. For all of them,



(a) (b)

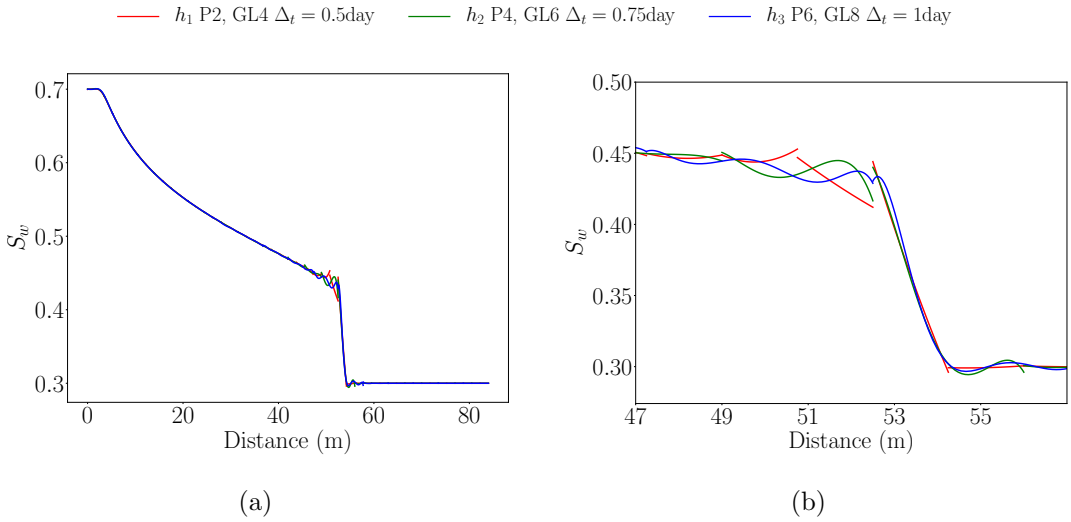
Figure 5.4: Plot over the line $y = 0$ at 30 days using a mesh composed of quadrilateral elements of polynomial degree $P = 4$ with size $h_e = 7$ meters, and using several time integration schemes with $\Delta t = 2$ days: a) water saturation profile, and b) zoom around the sharp front.

we use the same time step of $\Delta t = 2$ days. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 4$, see Equation (5.15).

Figure 5.4 shows the water saturation profile over the line $y = 0$ for the different time integration schemes at time 30 days. The smoothest saturation profile is obtained with the backward Euler since it introduces the highest amount of dissipation error, see Figure 5.4(a). As we use more accurate time integration schemes, the dissipation error is reduced and the waterfront becomes more vertical. Note that the spatial discretization is the same for all the time integration schemes. Therefore, as we increase the order of the temporal scheme, we obtain more discontinuities between elements and higher oscillations since high-order temporal schemes do not dissipate the errors introduced by the spatial discretization.

5.6.4 High-order spatial and temporal discretizations analysis

In this example, we analyze the advantages of combining high-order discretizations for space and time. We consider the same rectangular domain, boundary conditions and material parameters of Example 5.6.2.



(a) (b)

Figure 5.5: Plot over the line $y = 0$ at 30 days using three different spatial and temporal discretizations with the same resolution: a) water saturation profile and b) zoom around the sharp front of the water saturation profile.

We perform a waterflooding simulation with different element sizes, polynomial degrees and time integration schemes while keeping the same space-time resolution. That is, the total number of spatial and temporal unknowns is the same in all cases. To this end, we use three different spatial and temporal discretizations. In the first one, we use quadrilateral elements of polynomial degree $P = 2$ with size $h_1 = 1.75$ meters, and GL4 scheme with $\Delta t = 0.5$ days. In the second one, we use quadrilateral elements of polynomial degree $P = 4$ with size $h_2 = 3.5$ meters, and GL6 scheme with $\Delta t = 0.75$ days. In the third one, we use quadrilateral elements of polynomial degree $P = 6$ with size $h_3 = 5.25$ meters, and GL8 scheme with $\Delta t = 1$ day. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 2$ in Equation (5.15). The amount of artificial viscosity is the same in all the cases since the resolution of the spatial discretization is the same.

Figure 5.5 shows the plot over the line $y = 0$ at time 30 days of the water saturation obtained with the selected spatial and time discretizations. We observe that the sharp front is more vertical when high-order spatial and temporal discretizations are used. Moreover, as we increase the order of the spatial and temporal discretizations, the discontinuities between elements are reduced.

We highlight that the result of this example shows that high-order spatial and temporal discretizations obtain more accurate results than low-order discretizations

with the same resolution.

5.6.5 Accuracy of the time integration

In this example, we compare the accuracy and the computational cost of the midpoint and GL8 schemes. We consider the same rectangular domain, boundary conditions and material parameters of Example 5.6.2. First, we discretize the domain using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with size $h_e = 16.8$ meters. Using this mesh, we perform two simulations using the GL8 and the midpoint schemes with the same time step, $\Delta t = 4$ days. For both cases, we compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 6.4$, see Equation (5.15).

Figure 5.6 compares the water saturation approximation over the line $y = 0$ at time 32 days using the GL8 and midpoint schemes. The waterfront is more vertical when we use the GL8 scheme than when we use the midpoint scheme with the same time step. This illustrates that the midpoint scheme introduces more dissipation error than the GL8 scheme with the same time step. If we keep the solution obtained with the GL8 scheme as reference, we need to reduce five times the time step of the midpoint scheme ($\Delta t = 4/5 = 0.8$ days) to obtain a saturation profile with similar dissipation error. This leads to 8 number of time steps for the GL8 scheme and 40 number of time steps for the midpoint scheme.

Second, we reduce by half the element size, using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with size $h_e = 8.4$ meters. Therefore, we also divide by two the time step of the GL8 and the modified one for the midpoint. That is, we use $\Delta t = 2$ days for the GL8 scheme and $\Delta t = 0.8/2 = 0.4$ days for the midpoint. Since we have increased the resolution by a factor of two we reduce by a half the value of ϵ_0 parameter. Thus, we compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 3.2$ in Equation (5.15).

Figure 5.7 compares the saturation solution over the line $y = 0$ at time 32 days computed with the GL8 and midpoint schemes. The waterfront is more vertical when we use the GL8 scheme. The midpoint scheme introduces more dissipation error in the solution than the GL8 scheme because the time step for the midpoint is too large for this mesh. To obtain a similar dissipation error with both time integration schemes, we need to reduce the modified time step of the midpoint scheme sixteen times. That is $\Delta t = 0.8/16 = 0.05$ days. In this particular case, there is a factor

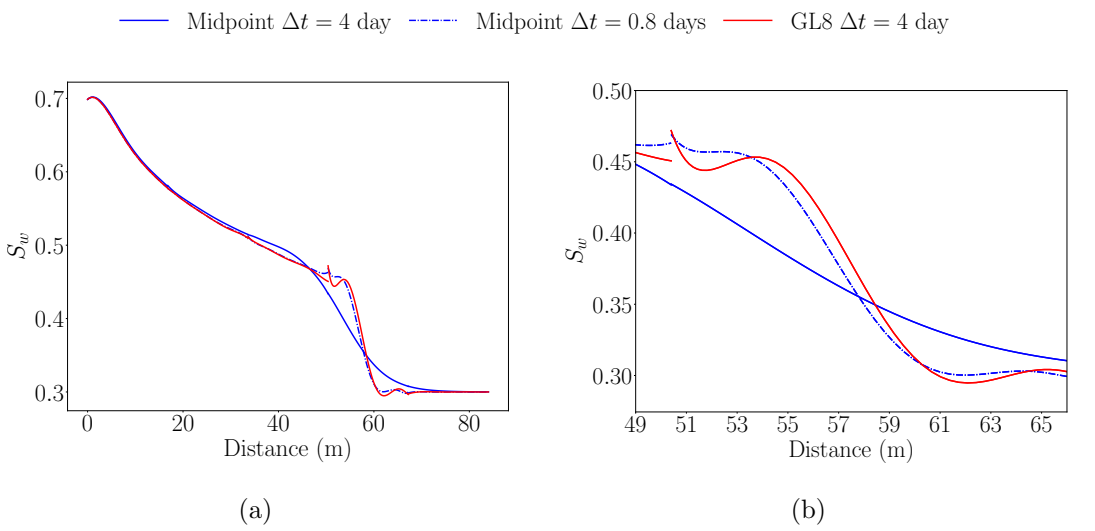


Figure 5.6: Plot over the line $y = 0$ at 32 days using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with $h_e = 16.8$ meters, and using the midpoint scheme with $\Delta t = 4$ and $\Delta t = 0.8$ days and GL8 scheme with $\Delta t = 4$: a) water saturation profile, and b) zoom around the sharp front.

of 40 between the number of time steps of the GL8 scheme ($\Delta t = 2$ days) and the midpoint scheme ($\Delta t = 0.05$ days). This leads to 16 number of time steps for the GL8 scheme and 640 number of time steps for the midpoint scheme.

It is important to highlight that once both methods obtain similar temporal accuracy, each time the GL8 time step is reduced by a factor α, the midpoint time step has to be reduced by a factor of α^z with $z = 8/2 = 4$. Therefore, to achieve the same temporal error between both schemes, the ratio between the number of time steps increases exponentially. Thus, high-order temporal schemes may reduce the computational cost because exponentially larger time steps can be used and exponentially less non-linear problems have to be solved.

5.6.6 Waterflooding through an heterogeneous material with obstacles

In this example, we simulate a case of waterflooding technique through a domain with two different material, see Figure 5.8(a). We consider a square domain $\Omega = (0, 100) \times (0, 100)$ with five circular obstacles of radius of 5 meters, located at $(25, 25)$, $(25, 50)$, $(25, 75)$, $(75, 37.5)$, $(75, 67.5)$ meters. The left side of the square is the injector

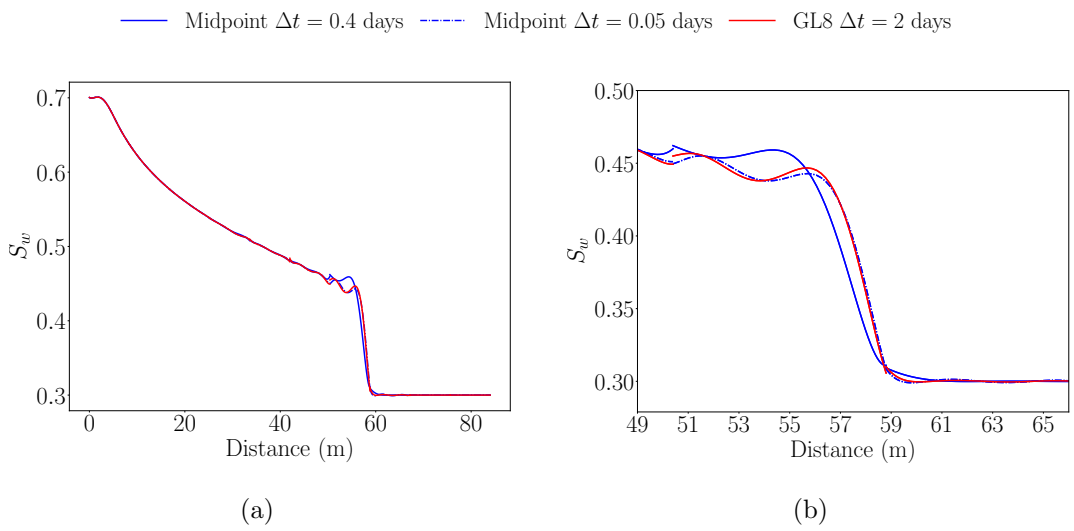


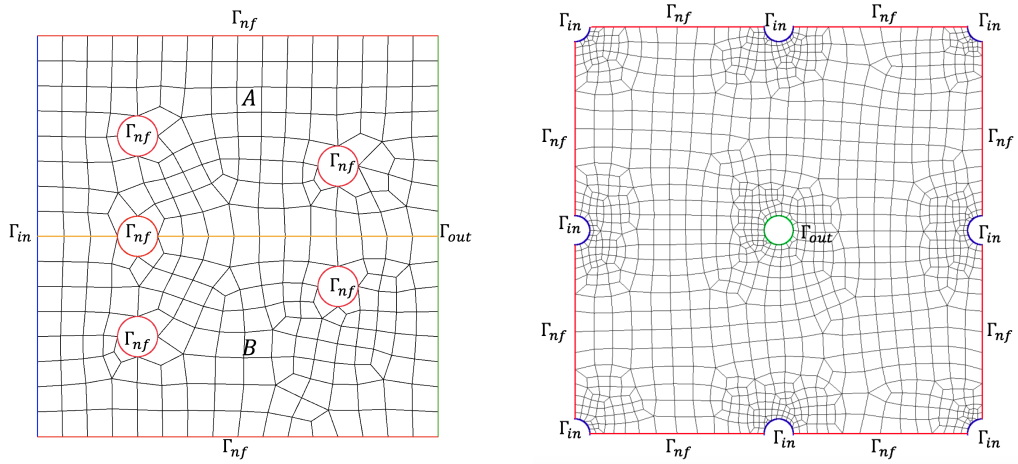
Figure 5.7: Plot over the line $y = 0$ at 32 days using a mesh composed of quadrilateral elements of polynomial degreee $P = 6$ with $h_e = 8.4$ meters, and using the midpoint scheme with $\Delta t = 0.4$ and $\Delta t = 0.05$ days and GL8 scheme with $\Delta t = 2$ days: a) water saturation profile, and b) zoom around the sharp front.

well, Γ_{in}, and the right side is the extractor well, Γ_{out}. The rest of the boundary is defined as no-flow, Γ_{nf}. We prescribe the same boundary conditions and the Brooks-Corey parameters as in Example 5.6.2.

The domain is composed of two materials with different permeability and porosity values. The upper half region is characterized by $\mathbf{K}_A = 5 \cdot 10^{-12}\mathbf{I}$ m^2 and $\phi_A = 0.4$, and the lower half region by $\mathbf{K}_B = 10^{-12}\mathbf{I}$ m^2 and $\phi_B = 0.2$. The viscosity for the water and oil phases are $\mu_w = 0.001$ Pa $\cdot$ s and $\mu_o = 0.012$ Pa $\cdot$ s, respectively.

We discretize the domain using 283 unstructured quadrilateral elements of polynomial degree four (4692 nodes), see Figure 5.8(a). To perform the time integration, we use the GL6 scheme with a time step $\Delta t = 12$ hours. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 4.8$ of Equation (5.15).

Figure 5.9 presents the water saturation field at times $t = 11, 15, 21$ and 35 days. Initially, the oil saturates homogeneously the porous media. Afterwards, water is injected along the inflow boundary and mobilizes the oil towards the outflow boundary. Water moves faster within the upper region, where the permeability is higher than in the bottom region. Furthermore, the water leaks from the upper half region to the lower half region. Note that, the added articial viscosity allows performing the simulation since non-physical saturation values are avoided and a numerical solution



(a) (b)

Figure 5.8: Mesh and boundary conditions distributions for the simulation of the water flooding through: a) heterogeneous domain with obstacles, b) a nine-spot pattern.

without oscillations is obtained.

Figure 5.10 shows the computed water pressure at times $t = 11, 15, 21$ and 35 days. We observe that the highest water pressure values are on the inflow boundary and the lowest on the outflow boundary. Moreover, at the left of the circular obstacles, the water pressure is higher than at the right. The pressure solution is smooth even when the domain contains obstacles.

Figure 5.11 plots the magnitude of Darcy water and oil velocities at times $t = 15$ days and $t = 35$ days. Note that Darcy velocities are higher in the upper half region than in the lower half because the soil is more permeable. Also, the water phase moves faster than the oil phase, since it is less viscous. We also observe that as the fluid overcome the obstacles the magnitude of Darcy water and oil velocities are higher above and below the obstacles than in front and behind. The artificial viscosity term also allows obtaining smooth approximations of the velocities, since they depend on the saturation.

This example illustrates that the proposed methodology allows performing high-order accurate simulations in space and time of a waterflooding problem with heterogeneous materials using unstructured high-order curved meshes and sharp fronts not aligned with the mesh.

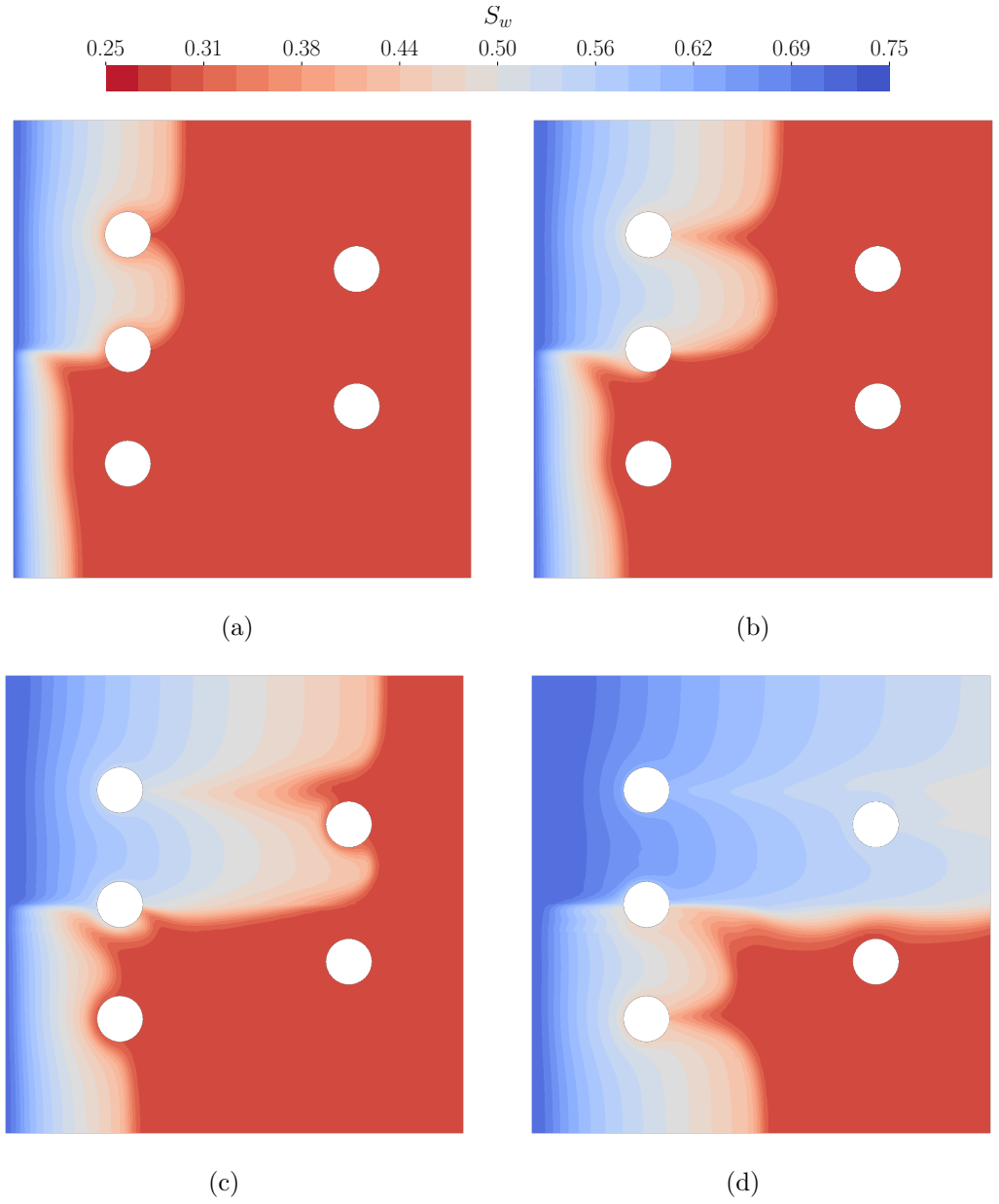


Figure 5.9: Water saturation approximation at time: a) 11 days, b) 15 days, c) 21 days, and d) 35 days.

5.6.7 Nine-spot pattern

In this example, we perform a waterflooding simulation for a nine spot pattern on a square domain, $\Omega = (0, 140) \times (0, 140)$ meters. This pattern has eight injection wells located at the vertices and the midpoint of the boundary edges, Γ_{in}, and one producer well located at the center of the domain, Γ_{out}. The rest of the boundary

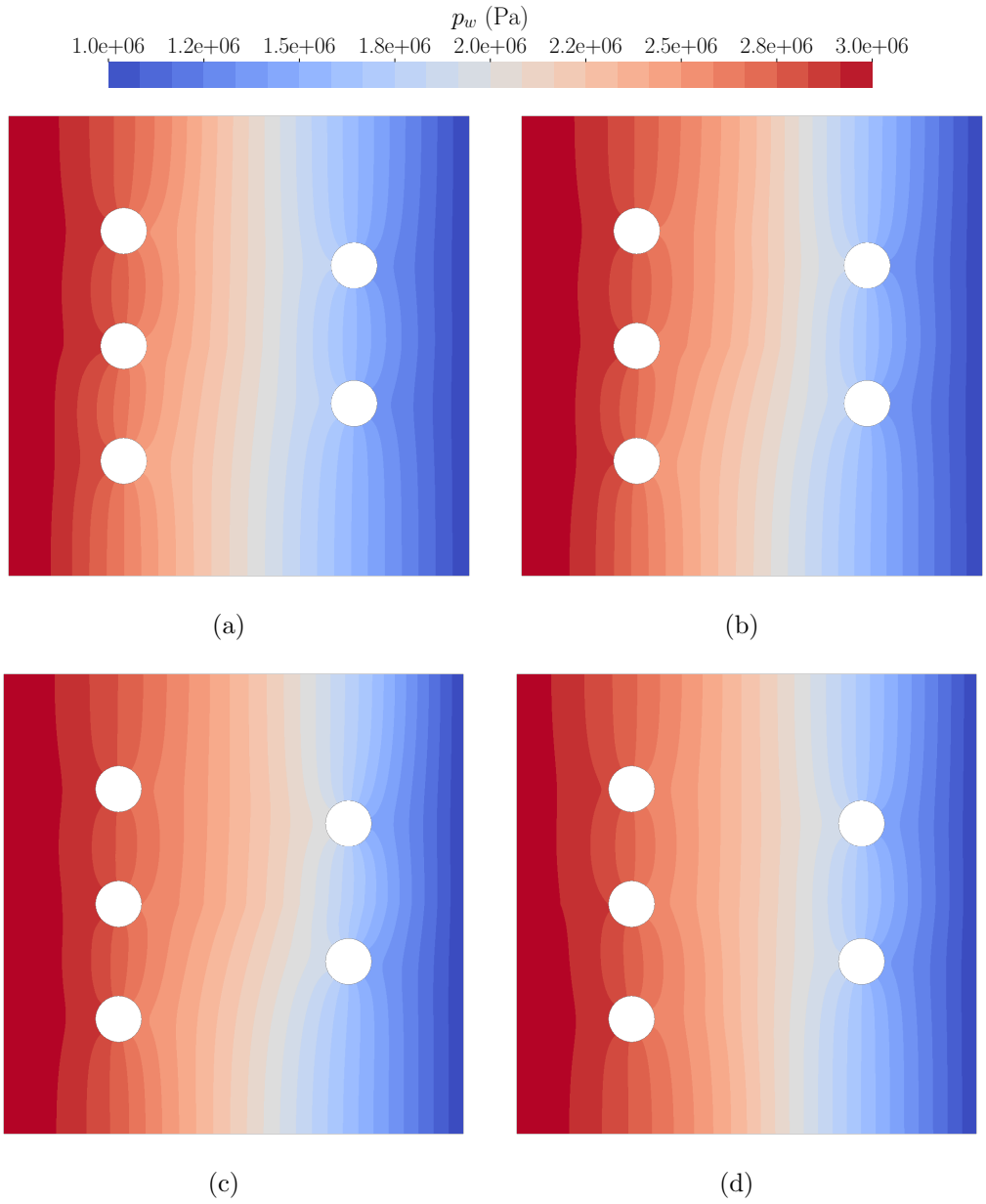


Figure 5.10: Water pressure approximation at time: a) 11 days, b) 15 days, c) 21 days, and d) 35 days.

is considered as no-flow, Γ_{nf}, see Figure 5.8(b). The radius of the wells is $r_w = 5$ m. We prescribe the same boundary conditions and Brooks-Corey parameters as in Example 5.6.2.

The soil permeability is $\mathbf{K} = 5 \cdot 10^{-12}\mathbf{I}$ m^2, the porosity is $\phi = 0.2$, and the viscosity for the water and oil phases are $\mu_w = 0.001$ Pa $\cdot$ s and $\mu_o = 0.012$ Pa $\cdot$ s,

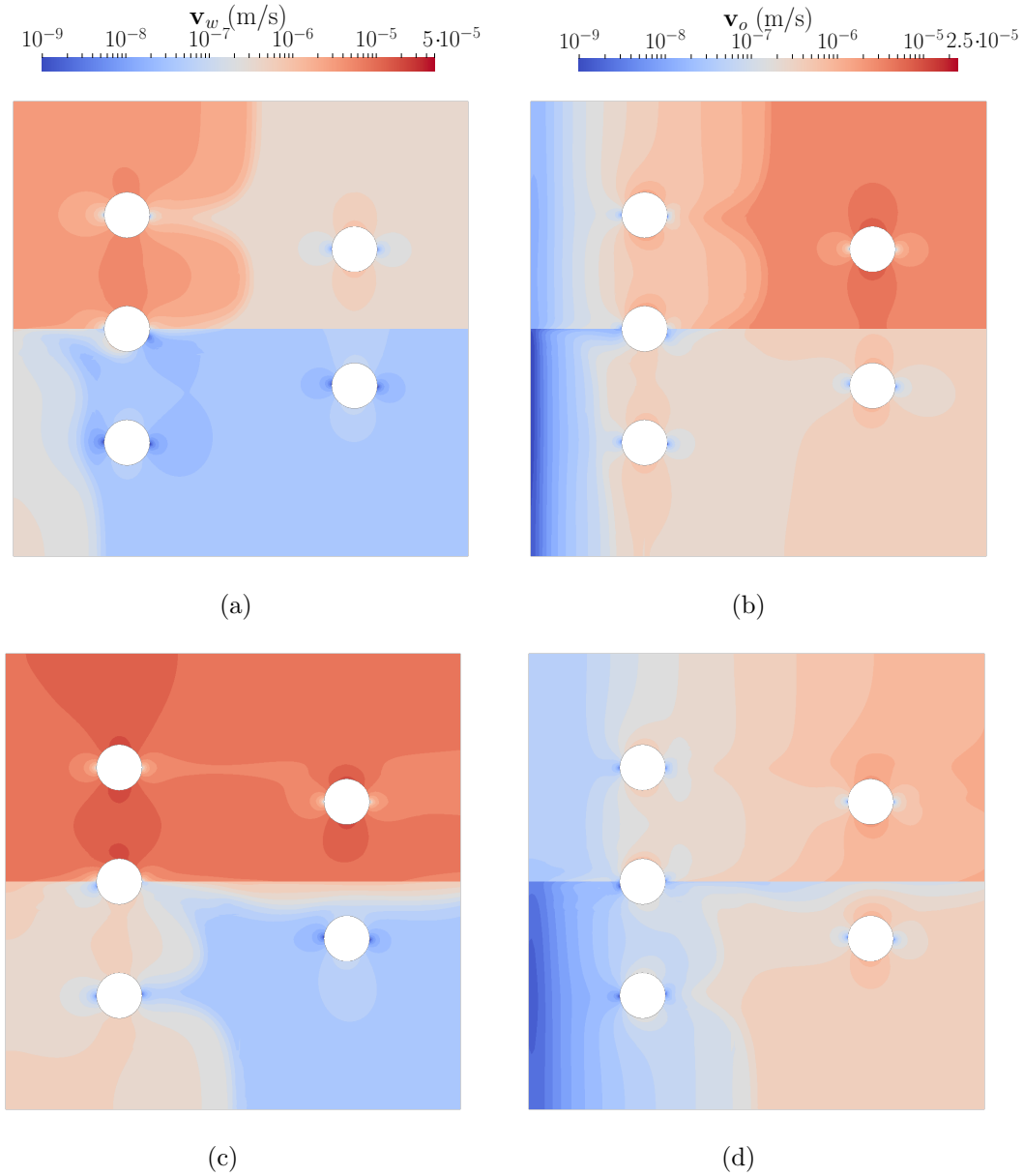


Figure 5.11: Magnitude of the Darcy velocities for the: a) water phase at time 15 days, b) oil phase at time 15 days, c) water phase at time 35 days and d) oil phase at time 35 days.

respectively.

We discretize the domain with 1114 non-constant size unstructured quadrilateral elements of polynomial degree four (18200 nodes), see Figure 5.8(b). To perform the time integration, we use the GL6 scheme with a time step $\Delta t = 6$ hours. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 0.78 h_e$ of Equation

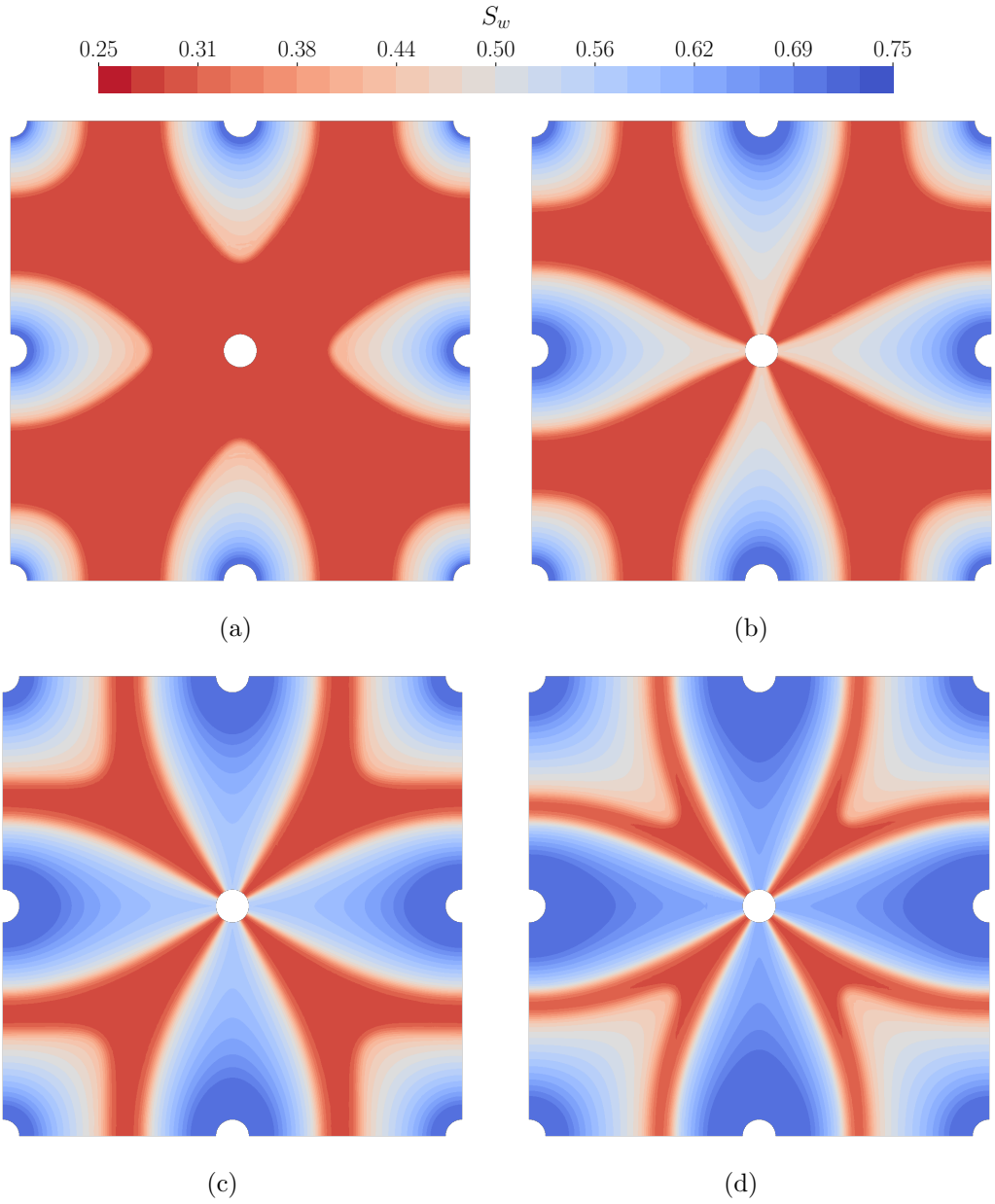


Figure 5.12: Water saturation approximation at time: a) 5 days, b) 7.5 days, c) 12.5 days and d) 17.25 days.

(5.15), being $h_e = \sqrt{\int_e 1 \mathrm{d}\Omega}$

Figure 5.12 presents the water saturation approximations at times $t = 5, 7.5, 12.5$ and 17.25 days. We inject water from the injectors wells, moving the oil to the pumping well at the centre, and occupying the space left by the oil phase. Thus, the water saturation increases from the injectors wells to the producer well.

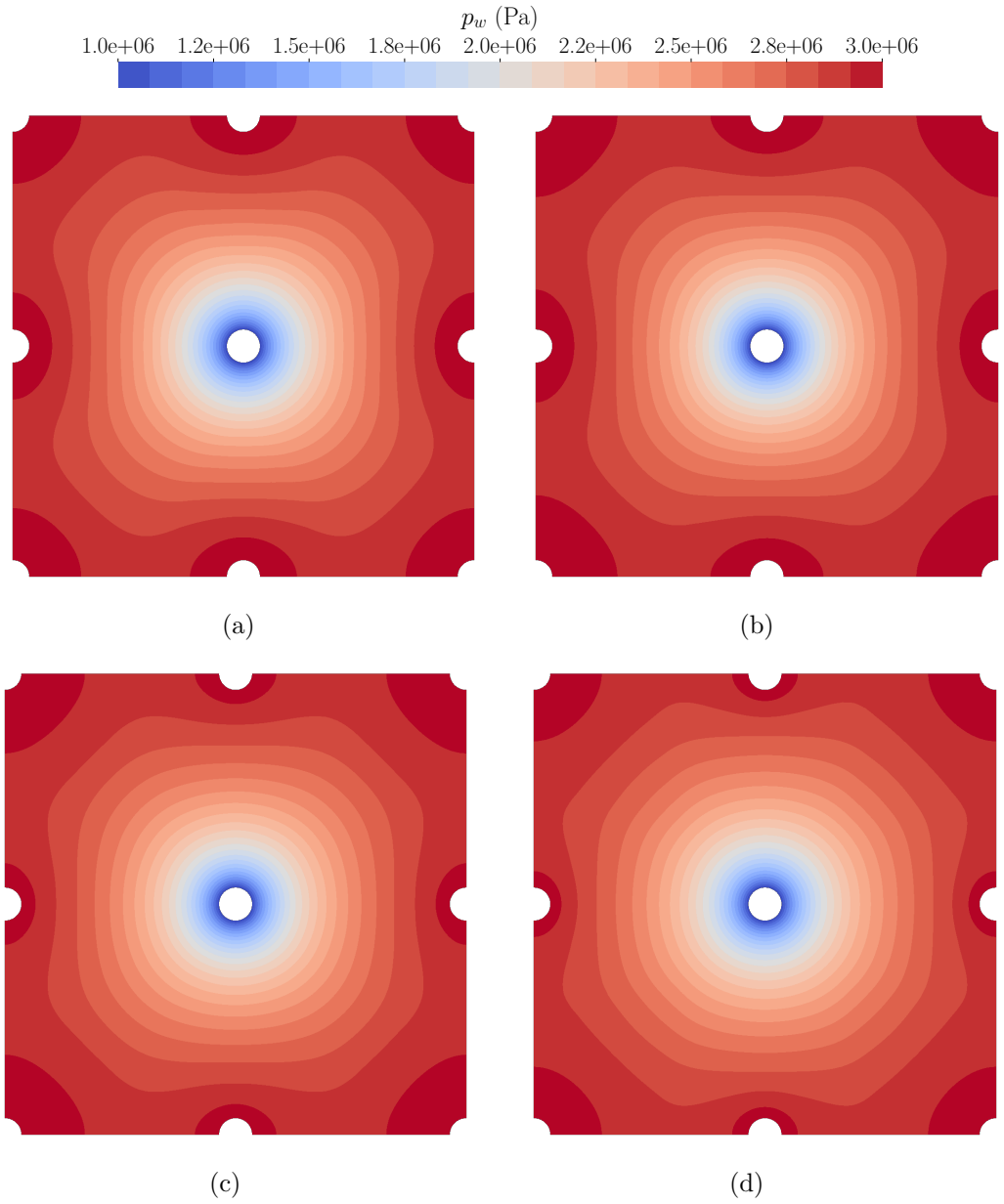


Figure 5.13: Water pressure approximation at time: a) 5 days,b) 7.5 days, c) 12.5 days and d) 17.25 days.

Figure 5.13 shows the water pressure field at times $t = 5, 7.5, 12.5$ and 17.25 days. As expected, the water pressure is higher at the injector wells and lower at the extractor well.

Figure 5.14 plots the magnitude of Darcy water and oil velocities at times 7.5 and 17.25 days. When the waterfront reaches the extractor well, both phases, oil and

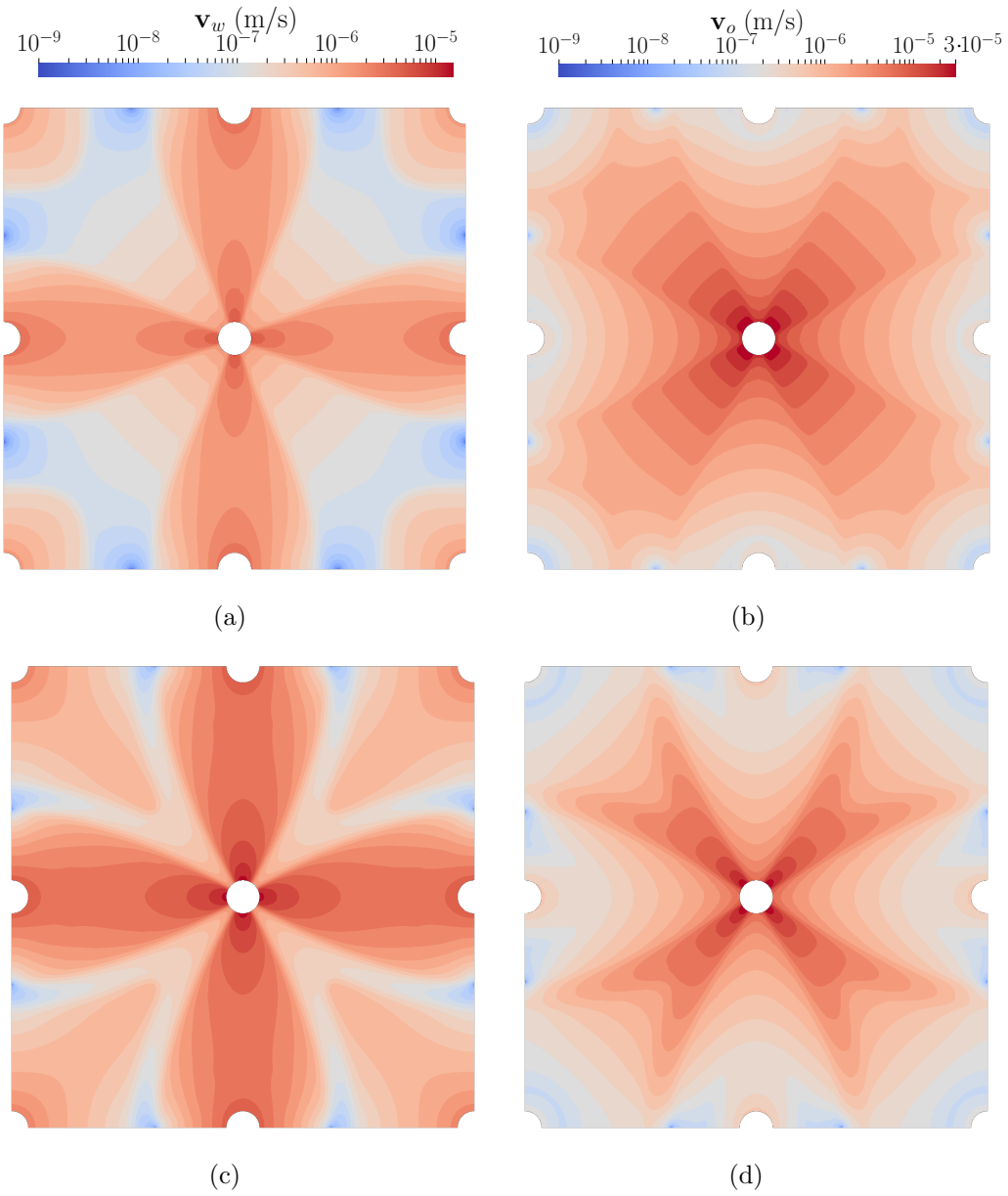


Figure 5.14: Magnitude of Darcy velocities for the: a) water phase at time 7.5 days,b) oil phase at time 7.5 days, c) water phase at time 17.25 days and d) oil phase at time 17.25 days.

water, are extracted. We observe that for all time steps the water velocity is higher than the oil velocity around the injector wells.

Note that the proposed high-order formulation and shock capturing technique can simulate several waterfronts (discontinuities) that interact between them. Moreover,

we can deal with high-order curved unstructured meshes of non-constant element size.

5.7 Concluding remarks

In this chapter, we have presented a memory-efficient high-order HDG formulation combined with high-order fully implicit time integration schemes for two-phase flow through porous media problem. We have assumed that the fluids are immiscible and the fluids and the porous media are incompressible. We have set the water pressure and oil saturation as the main unknowns. We have proposed a fix-point iterative method that alternatively solves the oil saturation and the water pressure implicitly until convergence is achieved. At each iteration of the fix-point, we solve the saturation equation by using the Newton-Raphson method, and the pressure equation at each stage independently by solving a linear system. The proposed fix-point iterative method is memory-efficient in the sense that the saturation and pressure are not solved at the same time. Furthermore, the pressure system is solved at each stage separately, since the pressure at different stages is not coupled. We have also proposed a temporal integration scheme in terms of the oil saturation instead of its temporal derivative. In this way, we have obtained a better sparsity pattern of the Jacobian with less coupling between the stages (Pazner and Persson, 2017). Finally, to deal with the sharp fronts that can appear in the saturation profile, we have introduced an artificial viscosity term in the saturation equation. To detect the location of the sharp fronts, we have used the shock sensor introduced in Persson and Peraire (2006), but computed from the saturation solution and the post-processed saturation of HDG. The proposed shock sensor is computationally efficient since the post-processed saturation is computed in an element-wise manner. We have proposed to compute the shock sensor at each RK stage and therefore, we introduce a different amount of artificial viscosity at each RK stage. This allows tracking the waterfront as it moves along the different RK stages.

We have presented several examples to assess the capabilities of the proposed formulation and methodology. First, we have shown that the proposed formulation is high-accurate in both space and time by studying the convergence rates for all the variables in space-time. Second, we have analyzed the proposed artificial viscosity term. We have observed that it is necessary to accurately determine the amount of artificial viscosity to reduce the spurious oscillations at the vicinity of the shock with-

out changing the underlying physics. We have shown that high-order methods can be competitive in terms of accuracy. That is, for the same time and spatial resolution we have obtained more vertical waterfronts and fewer discontinuities between elements when high-order methods are used. Moreover, we have analyzed the temporal error introduced by the temporal integration schemes. Low-order temporal schemes introduce more dissipation error into the solution. This leads to a diffused waterfront that does not correspond to the physics of the problem. We have also shown that if high-accuracy is required, high-order methods have a lower computational cost than low-order ones. Specifically, low-order temporal schemes may need exponentially smaller time steps to obtain solutions with similar errors than high-order temporal schemes. Finally, we have tested the robustness of the proposed shock sensor with two examples of the waterflooding technique. These examples deal with several waterfronts that also interact between them, heterogeneous material properties and unstructured high-order curved meshes.

Chapter 6

Summary and future work

6.1 Summary

In this section, we summarize the four main scientific contributions of this dissertation:

1. **We have proposed a stable high-order HDG formulation for slightly compressible one-phase flow through porous media coupled with high-order DIRK schemes.** In Chapter 3, we have presented a high-order HDG formulation for slightly compressible one-phase flow. This problem is modeled using a non-linear convection-diffusion equation, with the convection driven by gravity. We ensure the stability of the proposed HDG formulation by deducing an analytical expression for the stabilization parameter. To this end, we have split the stabilization parameter into diffusive and convective parts. The diffusive part is selected according to the physical values of the problem, and we have used the Engquist-Osher flux monotone scheme for the convective part. These choices ensure the existence and uniqueness of the obtained approximation while providing a stable and convergent method. To achieve high-accuracy in space and time, we have performed the temporal discretization using high-order DIRK schemes. This leads to a non-linear system at each stage of the DIRK scheme that we have solved using the Newton-Raphson method. Moreover, we have introduced the stabilization parameter in Newtons solver since we can analytically compute its derivatives.

2. **We have proposed a memory-efficient high-order HDG formulation for incompressible and immiscible two-phase flow through porous media problem coupled with high-order DIRK schemes.** In Chapter 4, we have presented a high-order HDG formulation for two-phase flow problem combined with DIRK schemes. We have assumed that the fluids are immiscible and the fluids and the porous media are incompressible. We have set the water pressure and oil saturation as the main unknowns, which leads to a coupled system of two non-linear partial differential equations. To solve it, we have used a fix-point iterative method that alternatively solves the oil saturation and the water pressure unknowns implicitly at each RK stage until convergence is achieved. The proposed fix-point iterative method is memory-efficient in the sense that the saturation and pressure are not solved at the same time.

3. **We have proposed a memory-efficient high-order HDG formulation for incompressible and immiscible two-phase flow through porous media problem coupled with high-order fully implicit RK schemes.** In Chapter 5, we have extended the formulation presented in Chapter 4, and we have coupled the HDG formulation with high-order fully implicit RK schemes. These schemes allow using arbitrary large time steps and achieve high-order temporal accuracy with few stages. Moreover, we have rewritten the non-linear system to reduce the memory footprint of the proposed formulation. That is, we have propose the temporal integration scheme in terms of the oil saturation instead of its temporal derivative. In this way, a better sparsity pattern in the Jacobian matrix and less coupling between the stages is obtained (Pazner and Persson, 2017). Furthermore, we have adapted the proposed fix-point iterative method in Chapter 4, which also reduces memory consumption since it decouples the saturation and pressure equations.

4. **Developing a robust and efficient shock capturing methodology.** In Chapter 5, we have detected the location of the sharp fronts using the shock sensor introduced in Persson and Peraire (2006), but computed from the saturation solution and the post-processed saturation of the HDG formulation. The proposed shock sensor is computationally efficient since the post-processed saturation is computed in an element-wise manner. We have computed the shock sensor at each RK stage and therefore, we were able to introduce a dif-

ferent amount of artificial viscosity at each RK stage. This allows tracking the waterfront as it evolves in time.

6.1.1 Research dissemination

We have published the presented contributions in a scientific article and chapter of a book:

- Costa-Solé, A., Ruiz-Gironés, E., and Sarrate, J. (2019). An HDG formulation for incompressible and immiscible two-phase porous media flow problems. International Journal of Computational Fluid Dynamics, 33(4), 137 – 148.

- Costa-Solé, A., Ruiz-Gironés, E., and Sarrate, J. (2020). One-phase and two-phase flow simulation using high-order HDG and high-order diagonally implicit time integration schemes. Accepted in Applied Mathematics for Environmental Problems ICIAM 2019 SEMA SIMAI Springer Series.

There are two more scientific articles under review process:

- Costa-Solé, A., Ruiz-Gironés, E., and Sarrate, J. (2020). High-order hybridizable discontinuous Galerkin formulation for one-phase flow through porous media.

- Costa-Solé, A., Ruiz-Gironés, E., and Sarrate, J. (2020). High-order HDG formulation with fully implicit temporal schemes for the simulation of two-phase flow through porous media.

Moreover, all the scientific contributions of this PhD thesis and the most relevant results have been presented in the following national and international conferences:

- Costa-Solé A, Ruiz Girons E. and Sarrate J. Two-phase flow simulation using high-order HDG and fully implicit time integration. ICIAM. Valencia, Spain, 2019.

- Costa-Solé A, Ruiz Gironés E. and Sarrate J. An HDG formulation for two-phase flow through porous media simulations. WCCM, New York, USA, 2018.

- Costa-Solé A, Ruiz Gironés E. and Sarrate J. High-order HDG method for one phase flow simulation. ECCOMAS, Glasgow, UK, 2018.

- Costa-Solé A, Ruiz Gironés E. and Sarrate J. Hydridizable discontinuous Galerkin method for two phase flow problems. CMN, Valencia, Spain, 2017. (Presentation and Poster)

- Costa-Solé A, Ruiz Gironés E. and Sarrate J. High-order hydridizable discontinuous Galerkin formulation for two phase flow simulation. CEDYA + CMA, Cartagena, Spain, 2017.

We have been accepted to present in the following international conference:

- Costa-Solé A, Ruiz Gironés E. and Sarrate J. High-order HDG and fully implicit methods for two-phase flow simulation. WCCM-ECCOMAS, Paris, France, 2020.

6.2 Future work

Several aspects can be analyzed and improved in the near future:

- **Coupling the presented high-order HDG formulation for the slightly compressible one-phase flow with fully implicit Runge-Kutta schemes.** We plan to extend the work presented in Chapter 3 by coupling the HDG formulation with fully implicit RK schemes. In this way, arbitrary large time steps can be used achieving a high-order accuracy with few stages.

- **Increasing the computational efficiency of the two-phase flow high-order HDG formulation with both DIRK and fully implicit Runge-Kutta schemes.** To this end, we will use the Newton-Raphson method to solve the saturation and pressure unknowns at all the stages. While this can reduce the number of iterations of the non-linear solver, it will increase the memory footprint. For that reason, to store the system matrix we will require to parallelize the code. Moreover, we will need to use iterative linear solvers. Thus, we need to investigate efficient iterative linear solvers and appropriated pre-conditioners for fully implicit time integration schemes. Considering all these improvements, we will expect to simulate larger problems dealing with more heterogeneous and complex domains.

- **Using h-p adaptivity to increase accuracy while reducing oscillations and computational cost.** Using this technique, we will be able to use small low-order elements near the front and large high-order elements for the rest of the domain. Therefore, we expect increasing the accuracy of the obtained solutions while reducing the computational cost of the simulation. We will investigate an error indicator to detect the elements with more error. Afterwards, it will be necessary to determine which technique should be used, the h-adaptivity, the p-adaptivity, or both.

- **Further analysis of the stability for the proposed two-phase flow formulations.** We propose to obtain an analytical expression for the convective stabilization parameter in the saturation equation. This will ensure the existence and uniqueness of the obtained approximation, as well as the stability of the formulation. To this end, we will investigate different monotone flux schemes.

- **Extending the presented HDG formulations to other porous media flow models.** We are interested in developing high-order HDG formulations for other multiphase flow models, such as the black oil model and the compositional flow. Therefore, we will be able to simulate more realistic and complex scenarios. We also consider extending the developed formulations with other assumptions, such as miscibility and/or compressible rocks and phases.

Appendix A

Jacobian terms for one-phase flow formulation

In this Appendix, we write the Jacobian terms required to solve Equation (3.22) of the Newton-Raphson method. First, we deduce the partial derivatives of the numerical convective flux, $\hat{\mathbf{F}}_h \cdot \mathbf{n}$, Equation (3.11), respect to the pressure, p_h, and its trace, $\hat{p}_h$

$$\frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial p_j} = \begin{cases} \left(\dfrac{\mathbf{F}(p_h) \cdot \mathbf{n} - \hat{\mathbf{F}}_h \cdot \mathbf{n}}{(p_h - \hat{p}_h)} \right) N_j & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} \geq 0, \\ 0 & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} < 0, \end{cases} \tag{A.1}$$

$$\frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial \hat{p}_j} = \begin{cases} \left(\dfrac{-\mathbf{F}(\hat{p}_h) \cdot \mathbf{n} + \hat{\mathbf{F}}_h \cdot \mathbf{n}}{(p_h - \hat{p}_h)} \right) N_j^f & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} \geq 0, \\ \mathbf{F}'(\hat{p}_h) N_j^f & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} < 0, \end{cases} \tag{A.2}$$

where p_h and $\hat{p}_h$ are defined in Equations (3.14) and (3.16), respectively. Note that, for $p_h = \hat{p}_h$ the first equations of Equation (A.1) and (A.2) becomes

$$\lim_{p_h \to \hat{p}_h} \frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial p_j} = \lim_{p_h \to \hat{p}_h} \frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial \hat{p}_j} = \frac{1}{2} \mathbf{F}'(\hat{p}_h).$$

The partial derivatives of Equations (3.14) to (3.17) are

$$\frac{\partial p_h}{\partial p_i} = N_i, \qquad \frac{\partial p}{\partial \dot{p}_j} = N_j \Delta t a_{ii}, \qquad \frac{\partial \mathbf{q}_h}{\partial q_{ij}} = N_i \mathbf{e}_j, \qquad \frac{\partial \hat{p}_h}{\partial p_j} = N_j^f.$$

Finally, we obtain the partial derivative of $\hat{\mathbf{F}}_h \cdot \mathbf{n}$ respect to $\dot{p}_j$ as

$$\frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial \dot{p}_j} = \frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial p_j} \cdot \frac{\partial p_h}{\partial \dot{p}_i} = \Delta t a_{ii} \frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial p_j}.$$

Thus, the elemental Jacobian coefficients are

$$\begin{aligned}
(\mathbf{J}^e_{\dot{p}\dot{p}})_{i,j} &= \frac{\partial}{\partial \dot{p}_j}\Big((s\dot{p}_h, N_i)_e - (\mathbf{F}_h, \nabla N_i)_e + \langle \hat{\mathbf{F}}_h \cdot \mathbf{n}, N_i\rangle_{\partial e} + \langle \tau_{\text{diff}}\ p_i, N_i\rangle_{\partial e} - \langle \tau_{\text{diff}}\ \hat{p}_h, N_i\rangle_{\partial e}\Big) \\
&= \left(\left(c_t \Delta t a_{ii}(c_r \phi^{ref}\rho + \phi c_f \rho^{ref}) N_j, N_i\right)_e + \left(\phi\rho(p_i) c_t \frac{\partial \dot{p}}{\partial p_j}, N_i\right)_e\right) \\
&- \left(\Delta t a_{ii} \frac{\mathbf{K}\nabla z \mathrm{g}}{\mu} 2\rho c_f \rho^{ref} N_j, \nabla N_i\right)_e + \langle \frac{\partial}{\partial \dot{p}_j} \hat{\mathbf{F}}_h \cdot \mathbf{n}, N_i\rangle_{\partial e} \\
&+ \left\langle \Delta t a_{ii} \left(\frac{\mathbf{K}}{\mu l} c_f \rho^{ref} p_i + \tau_{\text{diff}}\right) N_j, N_i \right\rangle_{\partial e} + \left\langle \Delta t a_{ii} \frac{\mathbf{K}}{\mu l} c_f \rho^{ref} \hat{p}_h, N_i \right\rangle_{\partial e} \\
(\mathbf{J}^e_{\dot{p}q})_{i,jk} &= \frac{\partial}{\partial q_{jk}} \left(-(\mathbf{q}_h, \nabla N_i)_e + \langle \mathbf{q}_h \cdot \mathbf{n}, N_i\rangle_{\partial e}\right) = -(N_j \mathbf{e}_k, \nabla N_i)_e + \langle N_j \mathbf{n}_k, N_i\rangle_{\partial e} \\
(\mathbf{J}^{e,f}_{\dot{p}\hat{p}})_{i,j} &= \frac{\partial}{\partial \hat{p}_j}\Big(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n}, N_i\rangle_{\partial e} - \langle \tau_{\text{diff}}\ \hat{p}_h, N_i\rangle_{\partial e}\Big) = \langle \frac{\partial}{\partial \hat{p}_j} \hat{\mathbf{F}}_h \cdot \mathbf{n}, N_i\rangle_{\partial e} - \langle \tau_{\text{diff}} N^f_j, N_i\rangle_{\partial e} \\
(\mathbf{J}^e_{q\dot{p}})_{ij,k} &= \frac{\partial}{\partial \dot{p}_k}\left((p_i, \nabla \cdot N_i \mathbf{e}_j)_e + (\mathbf{A}^{-1}\mathbf{q}_h, N_i \mathbf{e}_j)_e\right) \\
&= \Delta t a_{ii} \left(\left(-\mathbf{q}_h \mu \mathbf{K}^{-1} \frac{\rho^{ref} c_f N_k}{\rho(p_i)^2}, N_i \mathbf{e}_j\right)_e - \left(N_k, \frac{\partial N_i}{\partial x_j}\right)_e\right) \\
(\mathbf{J}^e_{qq})_{ij,kl} &= \frac{\partial}{\partial q_{kl}}\left(\mathbf{A}^{-1}\mathbf{q}_h, N_i \mathbf{e}_j\right)_e = (\mathbf{A}^{-1} N_k \mathbf{e}_l, N_i \mathbf{e}_j)_e \\
(\mathbf{J}^{e,f}_{q\hat{p}})_{ij,k} &= \frac{\partial}{\partial \hat{p}_k} \langle \hat{p}_h, N_i \mathbf{e}_j \cdot \mathbf{n}\rangle_{\partial e} = \langle N^f_k, N_i \mathbf{n}_j\rangle_{\partial e} \\
(\mathbf{J}^{e,f}_{\hat{p}\dot{p}})_{i,j} &= \frac{\partial}{\partial \dot{p}_j}\Big(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n}, N^f_i\rangle_{\partial e} + \langle \tau_{\text{diff}}\ p_i, N^f_i\rangle_{\partial e} - \langle \tau_{\text{diff}}\ \hat{p}_h, N^f_i\rangle_{\partial e}\Big) \\
&= \langle \frac{\partial}{\partial \dot{p}_j} \hat{\mathbf{F}}_h \cdot \mathbf{n},\ N^f_i\rangle_{\partial e} + \left\langle \Delta t a_{ii} \left(\frac{\mathbf{K}}{\mu l} c_f \rho^{ref} p_i + \tau_{\text{diff}}\right) N_j, N_i \right\rangle_{\partial e} \\
&- \left\langle \Delta t a_{ii} \frac{\mathbf{K}}{\mu l} c_f \rho^{ref} N_j \hat{p}_h, N^f_i \right\rangle_{\partial e} \\
(\mathbf{J}^{e,f}_{\hat{p}q})_{i,jk} &= \frac{\partial}{\partial q_{jk}} \langle \mathbf{q}_h \cdot \mathbf{n},\ N^f_i\rangle_{\partial e} = \langle N_j \mathbf{n}_k, N^f_i\rangle_{\partial e} \\
(\mathbf{J}^{e,f}_{\hat{p}\hat{p}})_{i,j} &= \frac{\partial}{\partial \hat{p}_j}\Big(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n}, N^f_i\rangle_{\partial e} - \langle \tau_{\text{diff}}\ \hat{p}_h, N^f_i\rangle_{\partial e}\Big) = \langle \frac{\partial}{\partial \hat{p}_j} \hat{\mathbf{F}}_h \cdot \mathbf{n}, N^f_i\rangle_{\partial e} - \langle \tau_{\text{diff}}\ N^f_j, N^f_i\rangle_{\partial e}
\end{aligned}$$

Appendix B

Runge-Kutta schemes

In this Appendix, we summarize the Butcher's tables of the different time integration schemes used in this work, see Butcher (1964a,b, 2016); Montlaur et al. (2012); Kennedy and Carpenter (2016); Pazner and Persson (2017) for more details.

Table B.1: Butcher's table for the backward scheme.

1	1
	1

Table B.2: Butcher's table for the midpoint scheme.

1/2	1/2
	1

Table B.3: Butcher's table for the DIRK3-s3 scheme.

γ	γ		
$\frac{1+\gamma}{2}$	$\frac{1-\gamma}{2}$	γ	
1	$\frac{-6\gamma^2+16\gamma-1}{4}$	$\frac{6\gamma^2-20\gamma-1}{4}$	γ
	$\frac{-6\gamma^2+16\gamma-1}{4}$	$\frac{6\gamma^2-20\gamma-1}{4}$	γ

$\gamma = 0.4358665215$

Table B.4: Butcher's table for the 4-th Gauss-Legendre Runge-Kutta scheme.

$1/2 - 1/6\sqrt{3}$	$1/4$	$1/4 - 1/6\sqrt{3}$
$1/2 + 1/6\sqrt{3}$	$1/4 + 1/6\sqrt{3}$	$1/4$
	$1/2$	$1/2$

Table B.5: Butcher's table for the 6-th Gauss-Legendre Runge-Kutta scheme.

$1/2 - 1/10\sqrt{15}$	$5/36$	$2/9 - 1/15\sqrt{15}$	$5/36 - 1/30\sqrt{15}$
$1/2$	$1/36 + 1/24\sqrt{15}$	$2/9$	$5/36 - 1/24\sqrt{15}$
$1/2 + 1/10\sqrt{15}$	$5/36 + 1/30\sqrt{15}$	$2/9 + 1/15\sqrt{15}$	$5/36$
	$5/18$	$4/9$	$5/18$

Table B.6: Butcher's table for the 8-th Gauss-Legendre Runge-Kutta scheme.

$1/2 - w_2$	w_1	$w_1' - w_3 + w_4'$	$w_1' - w_3 - w_4'$	$w_1 - w_5$
$1/2 - w_2'$	$w_1 - w_3' + w_4$	w_1'	$w_1' - w_5'$	$w_1 - w_3' - w_4$
$1/2 + w_2'$	$w_1 + w_3' + w_4$	$w_1' + w_5'$	w_1'	$w_1 + w_3' - w_4$
$1/2 + w_2$	$w_1 + w_5$	$w_1' + w_3 + w_4$	$w_1' + w_3 - w_4'$	w_1
	$2w_1$	$2w_1'$	$2w_1'$	$2w_1$

$$
\begin{aligned}
w_1 &= 1/8 - \sqrt{30}/144 & w_1' &= 1/8 + \sqrt{30}/144 \\
w_2 &= 1/2(\sqrt{15 + 2\sqrt{30}})/35 & w_2' &= 1/2(\sqrt{15 - 2\sqrt{30}})/35 \\
w_3 &= w_2\left(1/6 + \sqrt{30}/24\right) & w_3' &= w_2'\left(1/6 - \sqrt{30}/24\right) \\
w_4 &= w_2\left(1/21 + 5\sqrt{30}/168\right) & w_4' &= w_2'\left(1/21 - 5\sqrt{30}/168\right) \\
w_5 &= w_2 - 2w_3 & w_5' &= w_2' - 2w_3'
\end{aligned}
$$

Table B.7: Butcher's table for the 5-th Radau IIA Runge-Kutta scheme.

$2/5 - \sqrt{6}/10$	$11/45 - 7\sqrt{6}/360$	$37/225 - 169\sqrt{6}/1800$	$-2/225 + \sqrt{6}/75$
$2/5 + \sqrt{6}/10$	$37/225 + 169\sqrt{6}/1800$	$11/45 + 7\sqrt{6}/360$	$-2/225 - \sqrt{6}/75$
1	$4/9 - \sqrt{6}/36$	$4/9 + \sqrt{6}/36$	$1/9$
	$4/9 - \sqrt{6}/36$	$4/9 + \sqrt{6}/36$	$1/9$

Bibliography

Abushaikha, A. S., D. V. Voskov, and H. A. Tchelepi (2017). Fully implicit mixed-hybrid finite-element discretization for general purpose subsurface reservoir simulation. *Journal of Computational Physics 346*, 514–538.

Arbogast, T., M. Juntunen, J. Pool, and M. F. Wheeler (2013). A discontinuous Galerkin method for two-phase flow in a porous medium enforcing H(div) velocity and continuous capillary pressure. *Computational Geosciences 17*(6), 1055–1078.

Aziz, K. and A. Settari (1979). *Petroleum reservoir simulation*. Chapman & Hall.

Babuska, I., B. A. Szabo, and I. N. Katz (1981). The p-version of the finite element method. *SIAM Journal on Numerical Analysis 18*(3), 515–545.

Badia, S. and R. Codina (2010). Stabilized continuous and discontinuous Galerkin techniques for Darcy flow. *Computer Methods in Applied Mechanics and Engineering 199*(25-28), 1654–1667.

Barrios, T. P., J. M. Cascón, and M. González (2015). A posteriori error analysis of an augmented mixed finite element method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering 283*, 909–922.

Bassi, F. and S. Rebay (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *Journal of Computational Physics 131*(2), 267–279.

Bastian, P. (2014). A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure. *Computational Geosciences 18*(5), 779–796.

Bear, J. and A. Verruijt (2012). *Modeling groundwater flow and pollution*, Volume 2. Springer Science & Business Media.

Brezzi, F., T. J. R. Hughes, L. D. Marini, and A. Masud (2005). Mixed discontinuous Galerkin methods for Darcy flow. *Journal of Scientific Computing 22*(1-3), 119–145.

Butcher, J. C. (1964a). Implicit Runge-Kutta processes. *Mathematics of Computation 18*(85), 50–64.

Butcher, J. C. (1964b). On Runge-Kutta processes of high order. *Journal of the Australian Mathematical Society 4*(2), 179–194.

Butcher, J. C. (1996). A history of Runge-Kutta methods. *Applied numerical mathematics 20*(3), 247–260.

Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons.

Casoni, E., J. Peraire, and A. Huerta (2013). One-dimensional shock-capturing for high-order discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids 71*(6), 737–755.

Chen, Z. (2000). Formulations and numerical methods of the black oil model in porous media. *SIAM Journal on Numerical Analysis 38*(2), 489–514.

Chen, Z., G. Huan, and B. Li (2004). An improved IMPES method for two-phase flow in porous media. *Transport in Porous Media 54*(3), 361–376.

Chen, Z., G. Huan, and Y. Ma (2006). *Computational methods for multiphase flows in porous media*, Volume 2. Siam.

Corey, A. T. (1964). Hydraulic properties of porous media. *Colorade State University, Hydraulic Papers* (3).

Donaldson, E., G. Chilingarian, and T. Yen (1985). *Enhanced oil recovery, I: fundamentals and analyses*, Volume 17A. Elsevier.

Donaldson, E., G. Chilingarian, and T. Yen (1989). *Enhanced oil recovery, II: processes and operations*, Volume 17B. Elsevier.

Donea, J. and A. Huerta (2003). *Finite element methods for flow problems*. John Wiley & Sons.

Epshteyn, Y. (2007). *HP primal discontinuous Galerkin finite element methods for two-phase flow in porous media*. Ph. D. thesis, University of Pittsburgh.

Epshteyn, Y. and B. Rivière (2006). On the solution of incompressible two-phase flow by a p-version discontinuous Galerkin method. *Communications in Numerical Methods in Engineering 22*(7), 741–751.

Epshteyn, Y. and B. Rivière (2007). Fully implicit discontinuous finite element methods for two-phase flow. *Applied Numerical Mathematics 57*(4), 383–401.

Ern, A., I. Mozolevski, and L. Schuh (2010). Discontinuous Galerkin approximation of two-phase flows in heterogeneous porous media with discontinuous capillary pressures. *Computer Methods in Applied Mechanics and Engineering 199*(23-24), 1491–1501.

Fabien, M. S., M. Knepley, and B. Rivière (2018). A hybridizable discontinuous Galerkin method for two-phase flow in heterogeneous porous media. *International Journal for Numerical Methods in Engineering 116*(3), 161–177.

Fabien, M. S., M. Knepley, and B. Rivière (2020). A high order hybridizable discontinuous Galerkin method for incompressible miscible displacement in heterogeneous media. *Results in Applied Mathematics*, 100089.

Farmakis, P. S., P. Tsoutsanis, and X. Nogueira (2020). WENO schemes on unstructured meshes using a relaxed a posteriori MOOD limiting approach. *Computer Methods in Applied Mechanics and Engineering 363*, 112921.

Fernández, P., A. Christophe, S. Terrana, N. C. Nguyen, and J. Peraire (2018). Hybridized discontinuous Galerkin methods for wave propagation. *Journal of Scientific Computing 77*(3), 1566–1604.

Gargallo-Peiró, A., X. Roca, J. Peraire, and J. Sarrate (2015). Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering 103*(5), 342–363.

Gargallo-Peiró, A., X. Roca, J. Peraire, and J. Sarrate (2016). A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization. *International Journal for Numerical Methods in Engineering 106*(13), 1100–1130. nme.5162.

Giorgiani, G., S. Fernández-Méndez, and A. Huerta (2013). Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. *International Journal for Numerical Methods in Fluids 72*(12), 1244–1262.

Giorgiani, G., S. Fernández-Méndez, and A. Huerta (2014). Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible navier–stokes equations. *Computers & Fluids 98*, 196–208.

Hou, J., J. Chen, S. Sun, and Z. Chen (2016). Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media. *Journal of Computational and Applied Mathematics 307*, 262–283.

Huang, H. and G. Scovazzi (2013). A high-order, fully coupled, upwind, compact discontinuous Galerkin method for modeling of viscous fingering in compressible porous media. *Computer Methods in Applied Mechanics and Engineering 263*, 169–187.

Huerta, A., A. Angeloski, X. Roca, and J. Peraire (2013). Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering 96*(9), 529–560.

Huerta, A., E. Casoni, and J. Peraire (2012). A simple shock-capturing technique for high-order discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids 69*(10), 1614–1632.

Hughes, T. J. R., A. Masud, and J. Wan (2006). A stabilized mixed discontinuous Galerkin method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering 195*(25-28), 3347–3381.

Hurtado, F., C. Maliska, and A. Da Silva (2006). A variable timestep strategy for accelerating the IMPES solution algorithm in reservoir simulation. In *Proceedings of the XXVII Iberian Latin American Congress on Computational Methods in Engineering. Belém, Brasil: UFPA*, pp. 14.

Jamei, M. and H. Ghafouri (2016). A novel discontinuous Galerkin model for two-phase flow in porous media using an improved IMPES method. *International Journal of Numerical Methods for Heat & Fluid Flow 26*(1), 284–306.

Jamei, M., A. Raeisi Isa Abadi, and I. Ahmadianfar (2019). A Lax-Wendroff IMPES scheme for a two-phase flow in porous media using interior penalty discontinuous Galerkin method. *Numerical Heat Transfer, Part B: Fundamentals 75*(5), 325–346.

Kennedy, C. A. and M. H. Carpenter (2016). Diagonally implicit Runge-Kutta methods for ordinary differential equations. A review. Technical Report NASA/TM-2016-219173, NASA.

Kirby, R. M., S. J. Sherwin, and B. Cockburn (2012). To CG or to HDG: a comparative study. *Journal of Scientific Computing 51*(1), 183–212.

Klieber, W. and B. Rivière (2006). Adaptive simulations of two-phase flow by discontinuous Galerkin methods. *Computer Methods in Applied Mechanics and Engineering 196*(1-3), 404–419.

Kou, J. and S. Sun (2010). A new treatment of capillarity to improve the stability of IMPES two-phase flow formulation. *Computers & Fluids 39*(10), 1923–1931.

LeVeque, R. J. (1992). *Numerical methods for conservation laws*. Springer Science & Business Media.

Li, J. and B. Rivière (2015). High order discontinuous Galerkin method for simulating miscible flooding in porous media. *Computational Geosciences 19*(6), 1251–1268.

Löhner, R. (2011). Error and work estimates for high-order elements. *International Journal for Numerical Methods in Fluids 67*(12), 2184–2188.

Löhner, R. (2013). Improved error and work estimates for high-order elements. *International Journal for Numerical Methods in Fluids 72*(11), 1207–1218.

Luo, Y., M. Feng, and Y. Xu (2011). A stabilized mixed discontinuous Galerkin method for the incompressible miscible displacement problem. *Boundary Value Problems 2011*(1), 48.

Masud, A. and T. J. R. Hughes (2002). A stabilized mixed finite element method for Darcy flow. *Computer Methods in Applied Mechanics and Engineering 191*(39-40), 4341–4370.

Montlaur, A., S. Fernández-Méndez, and A. Huerta (2008). Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *International Journal for Numerical Methods in Fluids 57*(9), 1071–1092.

Montlaur, A., S. Fernández-Méndez, and A. Huerta (2012). High-order implicit time integration for unsteady incompressible flows. *International Journal for Numerical Methods in Fluids 70*(5), 603–626.

Natvig, J. R. and K.-A. Lie (2008). Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. *Journal of Computational Physics 227*(24), 10108–10124.

Nguyen, N. C., J. Peraire, and B. Cockburn (2009a). An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics 228*(9), 3232–3254.

Nguyen, N. C., J. Peraire, and B. Cockburn (2009b). An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics 228*(23), 8841–8855.

Nguyen, N. C., J. Peraire, and B. Cockburn (2011). High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics 230*(10), 3695–3718.

Nguyen, N. C., X. Roca, D. Moro, and J. Peraire (2013). A hybridized multiscale discontinuous Galerkin method for compressible flows. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, pp. 689.

Paipuri, M., S. Fernández-Méndez, and C. Tiago (2018). Comparison of high-order continuous and hybridizable discontinuous Galerkin methods for incompressible fluid flow problems. *Mathematics and Computers in Simulation 153*, 35–58.

Pazner, W. and P.-O. Persson (2017). Stage-parallel fully implicit Runge-Kutta solvers for discontinuous Galerkin fluid simulations. *Journal of Computational Physics 335*, 700–717.

Persson, P.-O. and J. Peraire (2006). Sub-cell shock capturing for discontinuous Galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, pp. 112.

Qiu, J. and C.-W. Shu (2005). Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing 26*(3), 907–929.

Radu, F. A., J. M. Nordbotten, I. S. Pop, and K. Kumar (2015). A robust linearization scheme for finite volume based discretizations for simulation of two-phase flow in porous media. *Journal of Computational and Applied Mathematics 289*, 134–141.

Rivière, B. (2005). Analysis of a discontinuous finite element method for the coupled Stokes and Darcy problems. *Journal of Scientific Computing 22*(1), 479–500.

Rivière, B. and M. F. Wheeler (2002). Discontinuous Galerkin methods for flow and transport problems in porous media. *Communications in Numerical Methods in Engineering 18*(1), 63–68.

Rivière, B., M. F. Wheeler, K. Banaś, et al. (2000). Part II. Discontinuous Galerkin method applied to a single phase flow in porous media. *Computational Geosciences 4*(4), 337–349.

Roca, X., N. C. Nguyen, and J. Peraire (2013). Scalable parallelization of the hybridized discontinuous Galerkin method for compressible flow. In *21st AIAA Computational Fluid Dynamics Conference*, pp. 2939.

Roca, X., E. Ruiz-Gironés, and J. Sarrate (2010). ez4u: mesh generation environment.

Ruiz-Gironés E.; Roca, X. and J. Sarrate (2016). High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation. *Computer-Aided Design 72*, 52 – 64.

Russell, T. F. and M. F. Wheeler (1983). Finite element and finite difference methods for continuous flows in porous media. In *The mathematics of reservoir simulation*, pp. 35–106. SIAM.

Salinas, P., D. Pavlidis, Z. Xie, C. Jacquemyn, Y. Melnikova, M. D. Jackson, and C. C. Pain (2017). Improving the robustness of the control volume finite element method with application to multiphase porous media flow. *International Journal for Numerical Methods in Fluids 85*(4), 235–246.

Salinas, P., D. Pavlidis, Z. Xie, H. Osman, C. C. Pain, and M. D. Jackson (2018). A discontinuous control volume finite element method for multi-phase flow in heterogeneous porous media. *Journal of Computational Physics 352*, 602–614.

Selley, R. and S. Sonnenberg (2014). *Elements of petroleum geology*. Academic Press.

Sevilla, R. and A. Huerta (2016). Tutorial on hybridizable discontinuous Galerkin (HDG) for second-order elliptic problems. In *Advanced Finite Element Technologies*, pp. 105–129. Springer.

Shu, C.-W. and S. Osher (1988). Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics 77*(2), 439–471.

Versteeg, H. K. and W. Malalasekera (2007). *An Introduction to computational fluid dynamics: the finite volume method*. Pearson education.

VonNeumann, J. and R. D. Richtmyer (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics 21*(3), 232–237.

Wang, Z., R. Fidkowski, Kand Abgrall, F. Bassi, D. Caraeni, A. Cary, H. D, R. Hartmann, K. Hillewaert, H. Huynh, et al. (2013). High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids 72*(8), 811–845.

Zhu, J., J. Qiu, C.-W. Shu, and M. Dumbser (2008). Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. *Journal of Computational Physics 227*(9), 4330–4353.

Zienkiewicz, O. C., R. L. Taylor, and J. Z. Zhu (2005). *The finite element method: its basis and fundamentals*. Elsevier.