ARTICLE TYPE

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

Albert Costa-Solé[1] | Eloi Ruiz-Gironés*[2] | Josep Sarrate[1]

[1]Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Spain
[2]Barcelona Supercomputing Center - BSC, Barcelona, Spain

Correspondence
*Eloi Ruiz-Gironés, Barcelona Supercomputing Center - BSC, Email: eloi.ruizgirones@bsc.es

Present Address
Jordi Girona 1-3, 08034 Barcelona, Spain

Summary

We present a memory-efficient high-order hybridizable discontinuous Galerkin (HDG) formulation coupled with high-order fully implicit Runge-Kutta schemes for immiscible and incompressible two-phase flow through porous media. To obtain the same high-order accuracy in space and time, we propose using high-order temporal schemes that allow using large time steps. Therefore, we require unconditionally stable temporal schemes for any combination of element size, polynomial degree and time step. Specifically, we use the Radau IIA and Gauss-Legendre schemes, which are unconditionally stable, achieve high-order accuracy with few stages, and do not suffer order reduction in this problem. 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 propose a fix-point iterative method to further reduce the memory consumption. The saturation solution may present sharp fronts. Thus, the high-order approximation may contain spurious oscillations. To reduce them, we introduce artificial viscosity. We detect the elements with high-oscillations using a computationally efficient shock sensor obtained from the saturation solution and the post-processed saturation of HDG. Finally, we present several examples to assess the capabilities of our formulation.

KEYWORDS:
Porous media, two-phase flow, high-order, hybridizable discontinuous Galerkin, fully implicit Runge-Kutta, artificial viscosity

1 | INTRODUCTION

Nowadays, several numerical methods are used to simulate multiphase flow through porous media in enhanced oil recovery. In this work, 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 to maintain the oil production rate[1]. To simulate this scenario, the immiscible two-phase flow through porous media model is widely used[2]. There are different formulations for this model depending on the selection of the main unknowns[2]. 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, one for the saturation and other for the pressure.

Several methods have been used to simulate two-phase flow, such as the finite element method[2], the finite volume method[2,3,4,5], mixed finite element methods[2,6,7], and discontinuous Galerkin methods[8,9,2,10,11,12,13,14,15,16,17]. Recently, many efforts have been

focused on applying high-order methods to these kind of problems due to their advantages [11,15,18,19,20]. 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 [21,22,23]. Hence, it has been shown that high-order spatial discretization methods can be more accurate than low-order ones for the same mesh resolution [23]. Moreover, it is also reported that high-order methods introduce less diffusion and dispersion errors in the solutions [21,22,23]. 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 [22,23,24,25].

Nevertheless, to obtain these advantages in unsteady problems, the temporal 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. The second one is to use high-order temporal schemes with larger 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 larger time steps may alleviate the computational cost while reducing the temporal dissipation and dispersion errors [26,27].

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 both have low-order memory footprint [28,29,30]. 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 [27]. Thus, if high accuracy is required, BDF methods still need small time steps and DIRK schemes need a high number of stages. In these cases, fully implicit RK schemes may be considered [27]. They are unconditionally stable and achieve high-order accuracy with few stages.

Combining high-order spatial discretization 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.

For non-smooth solutions (discontinuities or sharp fronts) spatial high-order methods will introduce oscillations at the vicinity of the discontinuity [31,32,33]. For 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.

In this paper, we propose a memory-efficient high-order hybridizable discontinuous Galerkin (HDG) formulation coupled with high-order fully implicit Runge-Kutta (RK) schemes for immiscible and incompressible two-phase flow through porous media. 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 oil saturation and water pressure variables and their corresponding fluxes of order $P+1$ in L^2-norm, being P the polynomial degree when the temporal discretization error is low enough [34,35,36,37,38]. Moreover, element-wise post-processing can be applied at chosen time steps to obtain a $P+2$ convergence rate for the oil saturation and water pressure [34,35,38]. Second, mass is conserved at the element level. This is an important feature when solving PDE's in conservative form. Third, the method can be hybridized in terms of the oil saturation and the water pressure traces, 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 [17,18,19,20].

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. These RK schemes are unconditionally stable for any combination of element size, polynomial degree and time-step [39]. In particular, we use the Gauss-Legendre (GL) and the Radau IIA 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. In this way, we obtain a better sparsity pattern in the Jacobian matrix and less coupling between the stages [27]. To further reduce the memory consumption, we propose a memory-efficient non-linear solver. Specifically, we perform a fix-point iterative method that alternatively solves the saturation and the pressure unknowns. That is, this method decouples the saturation and pressure systems. Therefore, we do not need to solve a non-linear system composed of both unknowns. Moreover,

since the pressure system does not contain temporal derivatives terms, it is solved sequentially at each Runge-Kutta stage given a saturation approximation.

If the saturation solution presents discontinuities, the high-order approximation may contain spurious oscillations[31,32,33]. To reduce these oscillations, we introduce local artificial viscosity in the saturation equation. The oscillations are detected using the shock sensor introduced in[31]. The main difference with[31] 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.

The outline of this paper is as follows. In Section 2, we describe the related work. In Section 3, we introduce the numerical model for the immiscible and incompressible two-phase flow through porous media problem. In Section 4, we deduce the corresponding HDG formulation. In Section 5, we couple the high-order HDG formulation with high-order fully implicit temporal discretization schemes. In Section 6, we detail the proposed non-linear solver. In Section 7, we state the local post-processing procedure. In Section 8, we specify the proposed shock-capturing method. In Section 9, we present several examples to asses the capabilities of the proposed formulation. Finally, in Section 10, we summarize the main contributions of this paper and we describe the issues that will be addressed in the future.

2 | RELATED WORK

Several spatial discretization methods have been successfully applied in reservoir simulation[2]. For instance, high-order continuous Galerkin methods can deal with unstructured meshes to capture the complexity of the subsurface configurations. Moreover, a static condensation procedure can be applied to reduce the number of unknowns of the linear systems and reduce the computational cost. In addition, 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[40,41]. Mixed finite element methods introduce the flux in the formulation, which is related to the gradient of the scalar variable. Both variables, the scalar and the flux have convergence rates in L^2-norm of $P+1$[2,6,7]. However, it is necessary to apply different stabilization techniques for each selection of approximation spaces[6]. 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 (DG) 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[8,9,2,10,11,13,12,14,16,15,17]. The stability, consistency and accuracy of these methods depend on a suitable choice of a numerical flux that depends on several parameters[42]. Also, the mass conservation is verified at the element level, which is an advantage when solving PDE's in conservative form. These methods can use polynomials of arbitrary degree, and therefore are high-order accurate. Moreover, there are DG formulations that allow solving only for the main unknown to increase the computational efficiency, which is known as primal formulation[11]. Recently, the hybridizable discontinuous Galerkin (HDG) method has been applied in two-phase flow through porous media[18,19,20]. It has all the advantages of the discontinuous Galerkin formulations. This method introduces the trace of the scalar variable as a new unknown. The stability is imposed through the continuity in the normal direction of a numerical flux that depends on a single stabilization parameter. Moreover, elemental post-processing can be applied to obtain a convergence rate for the scalar variables of $P+2$ in L^2-norm[34,35,36,37,38]. Therefore, the accuracy of the obtained solutions can be increased without hampering the computational cost. 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[39,43,44,28,29,27,30]. 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, we can use the implicit RK schemes[43]. 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[27]. 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[39,27]. However, the stage unknowns are all coupled, and therefore, they involve solving a non-linear system of equations that couples all the stages. Thus, fully implicit RK schemes have a higher memory footprint than explicit RK and DIRK methods.

For non-smooth solutions, high-order approximations may contain oscillations at the sharp fronts. The idea of the shock-capturing methods is to introduce additional dissipation near the sharp fronts to reduce these oscillations[31,32,33]. In this paper, we use the artificial viscosity method, which consists to introduce an elemental viscosity term into the governing equation[45]. This method requires a sensor to precisely detect the location of the sharp fronts. In the work of[31], the sensor is computed by a comparison of two solutions of the problem computed with a different polynomial degree, P and $P-1$. Moreover, the same authors relate the amount of artificial viscosity with the resolution of the mesh and two empirical parameters.

3 | NUMERICAL MODEL

The governing equations for two-phase flow through porous media are provided by the mass conservation and the Darcy law for each phase[2,46]:

$$\begin{cases} \dfrac{\partial(\phi\rho_\alpha S_\alpha)}{\partial t} + \boldsymbol{\nabla}\cdot\left(\rho_\alpha \mathbf{v}_\alpha\right) = \rho_\alpha f_\alpha, \\ \mathbf{v}_\alpha = -\lambda_\alpha \mathbf{K}\boldsymbol{\nabla} p_\alpha, \end{cases} \qquad \text{for} \quad \alpha = w, o$$

where w stands for the wetting phase (water), o stands for the non-wetting phase (oil), ϕ is the porosity of the media, ρ_α, f_α, $\mathbf{v}_\alpha$ S_α and p_α are the density, the source term, the velocity, the saturation and the pressure of phase α, 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.

We assume that both phases completely fill the voids of the soil, thus

$$S_w + S_o = 1.$$

There is a discontinuity in the pressure field due to the interface tension between phases called capillary pressure, p_c[2,46]. That is

$$p_c = p_o - p_w.$$

The capillary pressure, p_c, and the relative permeabilities of each phase, $k_{r\alpha}$, are related to the water or oil saturations by a physical model. In this work we use the Brooks-Corey model[47]:

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

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 depending on the selection of the main variables[2]. We use the oil saturation, S_o, and the water pressure, p_w, as main unknowns. According to[2,46], the total phase mobility, λ_t, and the total velocity, $\mathbf{v}_t$, are defined as

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

Let Ω be a domain and $T = (0, t_{end})$ a time interval. We consider that the boundary of Ω is divided in three disjointed parts such that $\partial\Omega = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma_{nf}$, 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 [10]. For the oil saturation:

$$\begin{cases} \phi\dfrac{\partial S_o}{\partial t} - \boldsymbol{\nabla}\cdot\left(\lambda_o\mathbf{K}(\boldsymbol{\nabla}p_c + \boldsymbol{\nabla}p_w)\right) = f_o & \forall(\mathbf{x},t)\in(\Omega,T), \\ S_o^{\Gamma_{in}} = g_{Ds}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ \left(\dfrac{\lambda_o\lambda_w}{\lambda_t}\mathbf{K}\boldsymbol{\nabla}p_c\right)\cdot\mathbf{n} = g_{Ns}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_o\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T), \\ S_o(\cdot,0) = S_i & \forall\mathbf{x}\in\Omega, \end{cases} \tag{2}$$

where g_{Ds}^{in} is the prescribed value of the saturation on the inflow boundary, and g_{Ns}^{out} is the prescribed value on the output boundary. For the water pressure:

$$\begin{cases} -\boldsymbol{\nabla}\cdot\left(\lambda_t\mathbf{K}\boldsymbol{\nabla}p_w + \lambda_o\mathbf{K}\boldsymbol{\nabla}p_c\right) = f_o + f_w & \forall(\mathbf{x},t)\in(\Omega,T), \\ p_w^{\Gamma_{in}} = g_{Dp}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ p_w^{\Gamma_{out}} = g_{Dp}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_t\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T), \end{cases} \tag{3}$$

where g_{Dp}^{in}, g_{Dp}^{out} are the prescribed values of the pressure on the inflow and outflow boundaries, respectively. We rewrite equations (2) and (3) as a system of first order PDEs by introducing the diffusive fluxes [34,35,37]:

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

Thus, the saturation system is

$$\begin{cases} \phi\dfrac{\partial S_o}{\partial t} + \boldsymbol{\nabla}\cdot\left(\mathbf{q}_s + \dfrac{\lambda_o}{\lambda_t}\mathbf{q}_p\right) = f_o & \forall(\mathbf{x},t)\in(\Omega,T), \\ \mathbf{q}_s + \lambda_o\mathbf{K}\boldsymbol{\nabla}p_c = \mathbf{0} & \forall(\mathbf{x},t)\in(\Omega,T), \\ S_o^{\Gamma_{in}} = g_{Ds}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ \left(\dfrac{\lambda_o\lambda_w}{\lambda_t}\mathbf{K}\boldsymbol{\nabla}p_c\right)\cdot\mathbf{n} = g_{Ns}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_o\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T), \\ S_o(\cdot,0) = S_i & \forall\mathbf{x}\in\Omega, \end{cases} \tag{4}$$

and the pressure system is

$$\begin{cases} \boldsymbol{\nabla}\cdot\left(\mathbf{q}_p + \mathbf{q}_s\right) = f_o + f_w & \forall(\mathbf{x},t)\in(\Omega,T), \\ \mathbf{q}_p + \lambda_t\mathbf{K}\boldsymbol{\nabla}p_w = \mathbf{0} & \forall(\mathbf{x},t)\in(\Omega,T), \\ p_w^{\Gamma_{in}} = g_{Dp}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ p_w^{\Gamma_{out}} = g_{Dp}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_t\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T). \end{cases} \tag{5}$$

4 | WEAK FORM

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:

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

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 $\mathbb{M}_h^P\left(g_D\right) =$

$\left\{\psi \in \mathbb{M}_h^P \mid \psi = \Pi(g_D) \text{ on } \Gamma_D\right\}$, where $\Pi(\cdot)$ is a projection operator to the space $\left\{\psi|_{\Gamma_D} \forall \psi \in \mathbb{M}_h^P\right\}$. In this work, we use a fixed polynomial degree for all the elements. We define the scalar products on the finite element spaces:

$$(u, v)_e = \int_e u\, v \, \mathrm{d}\Omega \qquad \forall u, v \in \mathbb{V}_h^P,$$

$$(\mathbf{q}, \mathbf{w})_e = \int_e \mathbf{q} \cdot \mathbf{w} \, \mathrm{d}\Omega \qquad \forall \mathbf{q}, \mathbf{w} \in \mathbf{W}_h^P,$$

$$\langle \lambda, \psi \rangle_{\partial e} = \int_{\partial e} \lambda\, \psi \, \mathrm{d}\Gamma \qquad \forall \lambda, \psi \in \mathbb{M}_h^P.$$

Using $\boldsymbol{\nabla} p_c = p_c' \boldsymbol{\nabla} S_o$, where p_c' is the derivative of the capillary pressure respect to the oil saturation, the HDG formulation for the oil saturation corresponding to Equation (4) seeks an approximation $(S_{o_h}, \mathbf{q}_{s_h}, \hat{S}_{oh}) \in \mathbb{V}_h^P \times \mathbf{W}_h^P \times \mathbb{M}_h^P(g_D)$ such that:

$$\sum_{e \in \mathrm{T}_h} \left(\left(\phi \frac{\partial S_{o_h}}{\partial t}, v \right)_e - \left(\mathbf{q}_{s_h} + \frac{\lambda_o}{\lambda_t} \mathbf{q}_{p_h}, \boldsymbol{\nabla} v \right)_e + \left\langle \left(\hat{\mathbf{q}}_{s_h} + \frac{\hat{\lambda}_o}{\hat{\lambda}_t} \hat{\mathbf{q}}_{p_h} \right) \cdot \mathbf{n}, v \right\rangle_{\partial \mathrm{T}_h} \right) = \sum_{e \in \mathrm{T}_h} \left(f_o, v \right)_e, \tag{6a}$$

$$\sum_{e \in \mathrm{T}_h} \left((\mathbf{A}_{s_h}^{-1} \mathbf{q}_{s_h}, \mathbf{w})_e - (S_{o_h}, \boldsymbol{\nabla} \cdot \mathbf{w})_e + \langle \hat{S}_{o_h}, \mathbf{w} \cdot \mathbf{n} \rangle_{\partial e} \right) = 0, \tag{6b}$$

$$\sum_{e \in \mathrm{T}_h} \sum_{\substack{f \in \partial e \\ f \notin \Gamma_{out}, \Gamma_{in}}} \left\langle \left(\hat{\mathbf{q}}_{s_h} + \frac{\hat{\lambda}_o}{\hat{\lambda}_t} \hat{\mathbf{q}}_{p_h} \right) \cdot \mathbf{n}, \psi \right\rangle_f = 0, \tag{6c}$$

$$\left\langle \frac{\lambda_o \lambda_w}{\lambda_t} p_c' \mathbf{K} \boldsymbol{\nabla} S_o \cdot \mathbf{n}, \psi \right\rangle_{\Gamma_{out}} = \langle g_{Ns}^{out}, \psi \rangle_{\Gamma_{out}} \tag{6d}$$

for all $(v, \mathbf{w}, \psi) \in \mathbb{V}_h^P \times \mathbf{W}_h^P \times \mathbb{M}_h^P(0)$, 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 $\mathbf{A}_s = \lambda_o p_c' \mathbf{K}$.

The HDG formulation for the water pressure corresponding to Equation (5) seeks an approximation $(p_{w_h}, \mathbf{q}_{p_h}, \hat{p}_{w_h}) \in \mathbb{V}_h^P \times \mathbf{W}_h^P \times \mathbb{M}_h^P(g_D)$ such that:

$$\sum_{e \in \mathrm{T}_h} \left(-(\mathbf{q}_{p_h} + \mathbf{q}_{s_h}, \boldsymbol{\nabla} v)_e + \langle (\hat{\mathbf{q}}_{p_h} + \hat{\mathbf{q}}_{s_h}) \cdot \mathbf{n}, v \rangle_{\partial e} \right) = \sum_{e \in \mathrm{T}_h} \left((f_o + f_w), v \right)_e, \tag{7a}$$

$$\sum_{e \in \mathrm{T}_h} \left((\mathbf{A}_{p_h}^{-1} \mathbf{q}_{p_h}, \mathbf{w})_e - (p_{w_h}, \boldsymbol{\nabla} \cdot \mathbf{w})_e + \langle \hat{p}_{w_h}, \mathbf{w} \cdot \mathbf{n} \rangle_{\partial e} \right) = 0, \tag{7b}$$

$$\sum_{e \in \mathrm{T}_h} \sum_{\substack{f \in \partial e \\ f \notin \Gamma_{out}, \Gamma_{in}}} \langle (\hat{\mathbf{q}}_{p_h} + \hat{\mathbf{q}}_{s_h}) \cdot \mathbf{n}, \psi \rangle_f = 0, \tag{7c}$$

for all $(v, \mathbf{w}, \psi) \in \mathbf{W}_h^P \times \mathbb{V}_h^P \times \mathbb{M}_h^P(0)$, where $\mathbf{A}_p = \lambda_t \mathbf{K}$, and $\hat{p}_{w_h}$ is the trace of the water pressure.

Equations (6c) and (7c) 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

$$\hat{\mathbf{q}}_{s_h} = \mathbf{q}_{s_h} + \tau_s (S_{o_h} - \hat{S}_{o_h}) \cdot \mathbf{n}, \tag{8a}$$

$$\hat{\mathbf{q}}_{p_h} = \mathbf{q}_{p_h} + \tau_p (p_{w_h} - \hat{p}_{w_h}) \cdot \mathbf{n}, \tag{8b}$$

respectively, where τ_s a stabilization function for the oil saturation, and τ_p a stabilization function for the water pressure. According to[34,35], we set the stabilization parameter, τ_s and τ_p, of Equations(8a) and (8b) respectively as

$$\tau_s = \frac{\hat{\lambda}_o p_c'}{l_s} \gamma_{\mathbf{K}}, \qquad \tau_p = \frac{\hat{\lambda}_t}{l_p} \gamma_{\mathbf{K}}, \tag{9}$$

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:

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

where $\partial \mathrm{T}^s_{h\Gamma_D}$ and $\partial \mathrm{T}^p_{h\Gamma_D}$ 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 $\mathbb{S}^P$, where N is the total number of element nodes, and let $\{N^f_i\}_{i=1,\dots,N_f}$ be a Lagrangian basis on the element faces, where N_f is the total number of nodes on a single face. 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

$$
\begin{aligned}
S_{o_h}(\mathbf{x},t) &= \sum_{i=1}^{N} S_i(t) N_i(\mathbf{x}), \qquad & p_{w_h}(\mathbf{x},t) &= \sum_{i=1}^{N} p_i(t) N_i(\mathbf{x}),\\
\mathbf{q}_{s_h}(\mathbf{x},t) &= \sum_{i=1}^{N}\sum_{j=1}^{N_{sd}} q_{si,j}(t) N_i \mathbf{e}_j(\mathbf{x}), \qquad & \mathbf{q}_{p_h}(\mathbf{x},t) &= \sum_{i=1}^{N}\sum_{j=1}^{N_{sd}} q_{pi,j}(t) N_i \mathbf{e}_j(\mathbf{x}),\\
\hat{S}_{o_h}(\mathbf{x},t) &= \sum_{i=1}^{N_f} \hat{S}_l(t) N^f_l(\mathbf{x}), \qquad & \hat{p}_{w_h}(\mathbf{x},t) &= \sum_{i=1}^{N_f} \hat{p}_l(t) N^f_l(\mathbf{x}).
\end{aligned}
\tag{10}
$$

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

$$
\dot{S}_{o_h}(\mathbf{x},t) = \sum_{i=1}^{N} \dot{S}_i(t) N_i(\mathbf{x}), \tag{11}
$$

where $\dot{S}_i(t) = d\dot{S}_i(t)/dt$. By inserting Equations (10) and (11) into Equations (6) and (7), we obtain a coupled system of first order differential algebraic equations (DAE).

The discrete problem consists of finding the coefficients $S_i(t), \dot{S}_i(t), q_{si,j}(t), p_i(t), q_{pi,j}(t)$, 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}, \boldsymbol{\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} \left(f_o, N_i \right)_e = 0\\
\left[\mathbf{R}_{\mathbf{q}_s}\right]_{i,j} \equiv & \sum_{e\in \mathrm{T}_h} \left((\mathbf{A}^{-1}_{s_h}\mathbf{q}_{s_h}, N_i\mathbf{e}_j)_e - (S_{o_h}, \boldsymbol{\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} \sum_{\substack{f\in\partial e\\ f\notin\Gamma_{out},\Gamma_{in}}} \left(\left\langle \left(\hat{\mathbf{q}}_{s_h} + \frac{\hat{\lambda}_o}{\hat{\lambda}_t}\hat{\mathbf{q}}_{p_h} \right)\cdot\mathbf{n}, N^f_l \right\rangle_f \right) + \left\langle \frac{\lambda_o\lambda_w}{\lambda_t} p'_c \mathbf{K}\boldsymbol{\nabla} S_o\cdot\mathbf{n}, N^f_l \right\rangle_{\Gamma_{out}} - \langle g^{out}_{Ns}, N^f_l\rangle_{\Gamma_{out}} = 0\\
\left[\mathbf{R}_{p_w}\right]_i \equiv & \sum_{e\in \mathrm{T}_h} \left(-(\mathbf{q}_{p_h} + \mathbf{q}_{s_h}, \boldsymbol{\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) = 0\\
& + \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} \left(f_o + f_w, N_i \right)_e = 0\\
\left[\mathbf{R}_{\mathbf{q}_p}\right]_{i,j} \equiv & \sum_{e\in \mathrm{T}_h} \left((\mathbf{A}^{-1}_{p_h}\mathbf{q}_{p_h}, N_i\mathbf{e}_j)_e - (p_{w_h}, \boldsymbol{\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} \sum_{\substack{f\in\partial e\\ f\notin\Gamma_{out},\Gamma_{in}}} \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^f_l\rangle_{\partial e} \right) = 0
\end{aligned}
\tag{12}
$$

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

5 | TIME DISCRETIZATION

We write the first order DAE system of Equation (12) in a compact form as

$$\mathbf{R}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) = \begin{bmatrix} \mathbf{R}_{S_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\hat{S}_o}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{p_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\mathbf{q}_p}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \\ \mathbf{R}_{\hat{p}_w}\left(t, \mathbf{S}_o, \dot{\mathbf{S}}_o, \mathbf{q}_s, \hat{\mathbf{S}}_o, \mathbf{p}_w, \mathbf{q}_p, \hat{\mathbf{p}}_w\right) \end{bmatrix} = \mathbf{0}, \tag{13}$$

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 (13), we use a fully implicit RK method. 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 RK stage. Thus, we compute the oil saturation at time $t^{n+1} = t^n + \Delta t$ as

$$\mathbf{S}_o^{n+1} = \mathbf{S}_o^n + \Delta t \sum_{i=1}^{s} b^i \dot{\mathbf{S}}_o^{n,i}, \tag{14}$$

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:

$$\mathbf{R}\left(t^{n,i}, \mathbf{S}_o^{n,i}, \dot{\mathbf{S}}_o^{n,i}, \mathbf{q}_s^{n,i}, \hat{\mathbf{S}}_o^{n,i}, \mathbf{p}_w^{n,i}, \mathbf{q}_p^{n,i}, \hat{\mathbf{p}}_w^{n,i}\right) = \mathbf{0}, \quad \text{for } i = 1, \dots, s, \tag{15}$$

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

$$\mathbf{S}_o^{n,i} = \mathbf{S}_o^n + \Delta t \sum_{j=1}^{s} a^{ij} \dot{\mathbf{S}}_o^{n,j}, \tag{16}$$

The parameters a^{ij}, b^i, c^i define the RK method, and are given by the Butcher's tables[39,43,28,29]:

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b} \end{array} \equiv \begin{array}{c|cccc} c^1 & a^{11} & a^{12} & \dots & a^{1s} \\ c^2 & a^{21} & a^{22} & \dots & a^{2s} \\ \vdots & \vdots & & \ddots & \vdots \\ c^s & a^{s1} & & \dots & a^{ss} \\ \hline & b^1 & b^2 & \dots & b^s \end{array} \tag{17}$$

Instead of solving Equation (15) 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, see details in[27]. To this end, we first rewrite Equation (16) as

$$\dot{\mathbf{S}}_o^{n,i} = \frac{1}{\Delta t} \sum_{j=1}^{s} \tilde{a}^{ij} \left(\mathbf{S}_o^{n,j} - \mathbf{S}_o^n\right), \tag{18}$$

where $\tilde{a}^{ij} = (\mathbf{A}^{-1})^{ij}$. Afterwards, we rewrite Equation (18) as

$$\dot{\mathbf{S}}_o^{n,i} = \frac{1}{\Delta t} \sum_{j=1}^{s} \tilde{a}^{ij} \mathbf{S}_o^{n,j} - \frac{\tilde{c}^i}{\Delta t} \mathbf{S}_o^n, \tag{19}$$

where $\tilde{c}^i = \sum_{j=1}^{s} \tilde{a}^{ij}$. Thus, inserting Equation (19) into Equation (15) we obtain the following non-linear algebraic equation:

$$\mathbf{R}\left(t^{n,i}, \mathbf{S}_o^{n,i}, \frac{1}{\Delta t} \sum_{j=1}^{s} \tilde{a}^{ij} \mathbf{S}_o^{n,j} - \frac{\tilde{c}^i}{\Delta t} \mathbf{S}_o^n, \mathbf{q}_s^{n,i}, \hat{\mathbf{S}}_o^{n,i}, \mathbf{p}_w^{n,i}, \mathbf{q}_p^{n,i}, \hat{\mathbf{p}}_w^{n,i}\right) = \mathbf{0}, \quad \text{for } i = 1, \dots, s. \tag{20}$$

Once the oil saturation is computed at all stages, we compute the oil saturation at next time step by inserting Equation (18) into Equation (14):

$$\mathbf{S}_o^{n+1} = \tilde{d} \mathbf{S}_o^n + \sum_{j=1}^{s} \tilde{b}^j \mathbf{S}_o^{n,j} \tag{21}$$

Algorithm 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.(22).
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.(23).
8: $l = l + 1$
9: **check convergence** using Eq.(24).
10: $\mathbf{S}_o^{n,i} = \mathbf{S}_o^{n,i,l}$, $\mathbf{p}_w^{n,i} = \mathbf{p}_w^{n,i,l}$, $\mathbf{q}_p^{n,i} = \mathbf{q}_p^{n,i,l}$, $\mathbf{q}_s^{n,i} = \mathbf{q}_s^{n,i,l}$, $\hat{\mathbf{p}}_w^{n,i} = \hat{\mathbf{p}}_w^{n,i,l}$ $\hat{\mathbf{S}}_o^{n,i} = \hat{\mathbf{S}}_o^{n,i,l}$, .
11: end

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

6 | NON-LINEAR SOLVER

To solve Equation (20) 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 1. Let l be the l-th iteration of the fix-point iterative method. Thus, we first solve Equation (20) 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 1. That is,

$$\mathbf{R}\Bigg(\underbrace{t^{n,i}}_{Data}, \underbrace{\mathbf{S}_o^{n,i,l+1}, \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\mathbf{S}_o^{n,j,l+1}}_{Unknowns} - \underbrace{\frac{\tilde{c}^i}{\Delta t}\mathbf{S}_o^n}_{Data}, \underbrace{\mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{Unknowns}, \underbrace{\mathbf{p}_w^{n,i,l}, \mathbf{q}_p^{n,i,l}, \hat{\mathbf{p}}_w^{n,i,l}}_{Data} \Bigg) = \mathbf{0}, \text{ for } i = 1, \dots, s. \tag{22}$$

Once the saturation unknowns at all stages are computed, we solve Equation (20) 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 1, by imposing:

$$\mathbf{R}\Bigg(\underbrace{t^{n,i}, \mathbf{S}_o^{n,i,l+1}, \frac{1}{\Delta t}\sum_{j=1}^{s} \tilde{a}^{ij}\mathbf{S}_o^{n,j,l+1} - \frac{\tilde{c}^i}{\Delta t}\mathbf{S}_o^n, \mathbf{q}_s^{n,i,l+1}, \hat{\mathbf{S}}_o^{n,i,l+1}}_{Data}, \underbrace{\mathbf{p}_w^{n,i,l+1}, \mathbf{q}_p^{n,i,l+1}, \hat{\mathbf{p}}_w^{n,i,l+1}}_{Unknowns} \Bigg) = \mathbf{0}, \text{ for } i = 1, \dots, s. \tag{23}$$

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

$$\begin{aligned}
&\frac{\|S_{o_h}^{n,i,l} - S_{o_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|S_{o_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{S_o}, && \frac{\|p_{w_h}^{n,i,l} - p_{w_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|p_{w_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{p_w},\\
&\frac{\|\mathbf{q}_{s_h}^{n,i,l} - \mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|\mathbf{q}_{s_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{\mathbf{q}_s}, && \frac{\|\mathbf{q}_{p_h}^{n,i,l} - \mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|\mathbf{q}_{p_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{\mathbf{q}_p},\\
&\frac{\|\hat{S}_{o_h}^{n,i,l} - \hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|\hat{S}_{o_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{\hat{S}_o}, && \frac{\|\hat{p}_{w_h}^{n,i,l} - \hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Omega)}}{\|\hat{p}_{w_h}^{n,i,l+1}\|_{L^2(\Omega)}} < \varepsilon_{\hat{p}_w},\\
&\|\mathbf{R}_{S_o}\|_2 < \varepsilon_{\mathbf{R}_{S_o}}, && \|\mathbf{R}_{p_w}\|_2 < \varepsilon_{\mathbf{R}_{p_w}},\\
&\|\mathbf{R}_{\mathbf{q}_s}\|_2 < \varepsilon_{\mathbf{R}_{\mathbf{q}_s}}, && \|\mathbf{R}_{\mathbf{q}_p}\|_2 < \varepsilon_{\mathbf{R}_{\mathbf{q}_p}},\\
&\|\mathbf{R}_{\hat{S}_o}\|_2 < \varepsilon_{\mathbf{R}_{\hat{S}_o}}, && \|\mathbf{R}_{\hat{p}_w}\|_2 < \varepsilon_{\mathbf{R}_{\hat{p}_w}},
\end{aligned} \tag{24}$$

where $\|\cdot\|_{L^2(\Omega)}$ is the norm of the $L^2(\Omega)$ 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 (22), 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.

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[34,35,37,38]. 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.

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.

6.1 | Saturation solver

We use the Newton-Raphson method to solve the saturation system, Equation (22). 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

$$\mathbf{u}^n = \begin{bmatrix} \mathbf{u}^{n,1} \\ \vdots \\ \mathbf{u}^{n,s} \end{bmatrix}, \quad \text{where} \qquad \mathbf{u}^{n,i} = \begin{bmatrix} \mathbf{S}_o^{n,i} \\ \mathbf{q}_s^{n,i} \\ \hat{\mathbf{S}}_o^{n,i} \end{bmatrix}, \quad \text{for } i = 1 \dots s. \tag{25}$$

Thus, the non-linear residual of the saturation is

$$\mathbf{F}(\mathbf{u}^n) = \begin{bmatrix} \mathbf{R}_S\left(t, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \mathbf{R}_{\hat{S}}\left(t, \mathbf{S}_o^{n,1}, \dot{\mathbf{S}}_o^{n,1}, \mathbf{q}_s^{n,1}, \hat{\mathbf{S}}_o^{n,1}, \mathbf{p}_w^{n,1}, \mathbf{q}_p^{n,1}, \hat{\mathbf{p}}_w^{n,1}\right) \\ \vdots \\ \mathbf{R}_S\left(t, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \\ \mathbf{R}_{\mathbf{q}_s}\left(t, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \\ \mathbf{R}_{\hat{S}}\left(t, \mathbf{S}_o^{n,s}, \dot{\mathbf{S}}_o^{n,s}, \mathbf{q}_s^{n,s}, \hat{\mathbf{S}}_o^{n,s}, \mathbf{p}_w^{n,s}, \mathbf{q}_p^{n,s}, \hat{\mathbf{p}}_w^{n,s}\right) \end{bmatrix}. \tag{26}$$

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

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

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

$$\mathbf{J}\left(\mathbf{u}_k^n\right)\delta\mathbf{u}_k^n = -\mathbf{F}\left(\mathbf{u}_k^n\right), \tag{27}$$

being $\mathbf{J}\left(\mathbf{u}_k^n\right)$ the Jacobian matrix of $\mathbf{F}$ evaluated at $\mathbf{u}_k^n$. 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:

$$\mathbf{J} = \begin{bmatrix} \mathbf{M}^{11} & \mathbf{M}^{12} & \dots & \mathbf{M}^{1s} \\ \mathbf{M}^{21} & \mathbf{M}^{22} & \dots & \mathbf{M}^{2s} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{M}^{s1} & \mathbf{M}^{s2} & \dots & \mathbf{M}^{ss} \end{bmatrix} + \begin{bmatrix} \mathbf{J}^{11} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}^{22} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}^{ss} \end{bmatrix}. \tag{28}$$

The first matrix in Equation (28) 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 that couples all the variables at stage i with all the variables at stage j of the temporal part. Note that, each $\mathbf{M}^{ij}$ has only one block that is not zero, which corresponds to the coupling of the saturation at stage i with the saturation of the stage j. As consequence,

all the remaining blocks of $\mathbf{M}^{ij}$ are zero. Therefore, $\mathbf{M}^{ij}$ for $i, j = 1 \dots s$ is

$$\mathbf{M}^{ij} = \begin{bmatrix} \mathbf{M}\frac{\tilde{a}^{ij}}{\Delta t} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{29}$$

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 (28) 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.

6.2 | Pressure solver

The pressure unknowns of the different stages are not coupled because the pressure equation does not contain a temporal part, see Equation (23) . 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 1, the pressure equation is linear because the unknowns related to the saturation are considered as parameters in the pressure equation.

7 | 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 $\mathbb{V}_h^{P+1}$ with convergence rate of $P + 2$ in the L^2-norm[34,35,37].

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

$$\begin{aligned} (\lambda_o \mathbf{K} p_c' \boldsymbol{\nabla} S^*_{o_h}, \boldsymbol{\nabla} v)_e &= -(\mathbf{q}_{s_h}, \boldsymbol{\nabla} v)_e \\ (S^*_{o_h}, 1)_e &= (S_{o_h}, 1)_e. \end{aligned} \tag{30}$$

Equation (30) is non-linear since λ_o and p_c' depend on $S^*_{o_h}$. To solve Equation (30), 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, $p^*_{w_h} \in \mathbb{V}_h^{P+1}$ on each element, e, and at all the stages of the RK scheme. Specifically, we solve

$$\begin{aligned} (\lambda_t \mathbf{K} \boldsymbol{\nabla} p^*_{w_h}, \boldsymbol{\nabla} v)_e &= -(\mathbf{q}_{p_h}, \boldsymbol{\nabla} v)_e \\ (p^*_{w_h}, 1)_e &= (p_{w_h}, 1)_e, \end{aligned} \tag{31}$$

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

8 | 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[45,31,32,33]. 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[45,31,32,33]. We identify the elements containing the sharp front using the shock sensor introduced in[31]. The main difference with[31] is that we compute the shock sensor from the saturation solution and the post-processed saturation of HDG, $S^*_{o_h}$, obtained in Equation (30). Therefore, the shock sensor is computed in an efficient manner as

$$S_e = \frac{(S_{o_h} - S^*_{o_h}, S_{o_h} - S^*_{o_h})_e}{(S^*_{o_h}, S^*_{o_h})_e}, \tag{32}$$

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 are computed 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[31], we define the artificial viscosity factor, ε, which is related to the resolution of the spatial discretization, as

$$\varepsilon = \begin{cases} 0 & \text{if } s_e < s_0 - \kappa, \\ \frac{\epsilon_0}{2}\left(1 + \sin\left(\frac{\pi(s_e - s_0}{2\kappa}\right)\right) & \text{if } s_0 - \kappa \leq s_e \leq s_o + \kappa, \\ \epsilon_0 & \text{if } s_e > s_0 + \kappa, \end{cases} \tag{33}$$

where $s_e = \log_{10} S_e$, $\epsilon_0 \simeq \frac{h_e}{P}$, 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:

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

In terms of the fist-order PDE's the system to be solved for the oil saturation is

$$\begin{cases} \phi\frac{\partial S_o}{\partial t} + \boldsymbol{\nabla}\cdot\left(\mathbf{q}_{s,\varepsilon} + \frac{\lambda_o}{\lambda_t}\mathbf{q}_p\right) = f_o & \forall(\mathbf{x},t)\in(\Omega,T), \\ \mathbf{q}_{s,\varepsilon} + \lambda_o\mathbf{K}_\varepsilon p_c'\boldsymbol{\nabla}S_o = \mathbf{0} & \forall(\mathbf{x},t)\in(\Omega,T), \\ S_o^{\Gamma_{in}} = g_{Ds}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ \left(\frac{\lambda_o\lambda_w}{\lambda_t}\mathbf{K}\boldsymbol{\nabla}p_c\right)\cdot\mathbf{n} = g_{Ns}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_o\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T), \\ S_o(\cdot,0) = S_i & \forall\mathbf{x}\in\Omega, \end{cases} \tag{34}$$

and for the water pressure is

$$\begin{cases} \boldsymbol{\nabla}\cdot\left(\mathbf{q}_p + \frac{\mathbf{q}_{s,\varepsilon}}{1+\varepsilon}\right) = f_o + f_w & \forall(\mathbf{x},t)\in(\Omega,T), \\ \mathbf{q}_p + \lambda_t\mathbf{K}\boldsymbol{\nabla}p_w = \mathbf{0} & \forall(\mathbf{x},t)\in(\Omega,T), \\ p_w^{\Gamma_{in}} = g_{Dp}^{in} & \forall(\mathbf{x},t)\in(\Gamma_{in},T), \\ p_w^{\Gamma_{out}} = g_{Dp}^{out} & \forall(\mathbf{x},t)\in(\Gamma_{out},T), \\ \mathbf{v}_t\cdot\mathbf{n} = 0 & \forall(\mathbf{x},t)\in(\Gamma_{nf},T). \end{cases} \tag{35}$$

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 (5) 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 (32). 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.

9 | EXAMPLES

In this section, we present seven examples to assess the capabilities of the proposed formulation. Specifically, example 9.1 shows the convergence rates for the obtained solutions of the proposed formulation. Example 9.2 analyses the effect of introducing artificial viscosity. Examples 9.3 to 9.5 focus on the accuracy of high-order spatial and temporal discretization methods of the proposed formulation. Finally, examples 9.6 and 9.7 present two cases of waterflooding simulation. The first one considers heterogeneous material properties and the second one analyses the nine-spot pattern. The Butcher tables of the used time integration schemes are detailed in [39,43,28,29].

9.1 | Convergence rate analysis

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

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

We approximate the space-time integration as

$$\begin{aligned}\int_0^{t_{end}} \int_\Omega \|u\left(\mathbf{x},t\right)\|^2 \mathrm{d}\Omega\, \mathrm{d}t &= \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h} \int_e \|u\left(\mathbf{x},t\right)\|^2 \mathrm{d}\Omega\, \mathrm{d}t = \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h} \int_{e^M} \|u\left(\xi_g,t\right)\|^2 |J_g| \mathrm{d}\boldsymbol{\xi}\, \mathrm{d}t \\ &\simeq \int_0^{t_{end}} \underbrace{\sum_{e\in \mathrm{T}_h} \sum_{g=1}^{N_g} \|u\left(\xi_g,t\right)\|^2 |J_g| \omega_g \mathrm{d}\boldsymbol{\xi}}_{f(t)}\, \mathrm{d}t = \int_0^{t_{end}} f\left(t\right) \mathrm{d}t,\end{aligned} \tag{36}$$

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,

$$\int_0^{t_{end}} f\left(t\right) \mathrm{d}t = \sum_n^{N_s} \int_{t^n}^{t^{n+1}} f\left(t\right) \mathrm{d}t \simeq \Delta t \sum_{n=1}^{N_s} \sum_{i=1}^{s} f\left(t^n + c^i \Delta t\right) b^i, \tag{37}$$

where N_s is the number of time steps. Afterwards, substituting Equation (37) into Equation (36) we obtain that the approximation of the space-time L^2-norm of a function is:

$$\begin{aligned}\|u\|^2_{L^2(\Omega,T)} &\simeq \int_0^{t_{end}} \sum_{e\in \mathrm{T}_h} \sum_{g=1}^{N_g} \|u\left(\xi_g,t\right)\|^2 |J_g| \omega_g \mathrm{d}\boldsymbol{\xi}\, \mathrm{d}t \\ &\simeq \sum_{n=1}^{N_s} \sum_{i=1}^{s} \sum_{e\in \mathrm{T}_h} \sum_{g=1}^{N_g} \|u\left(\xi_g,t^{n,i}\right)\|^2 |J_g| \omega_g b^i \Delta t\end{aligned} \tag{38}$$

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

$$\begin{aligned} E_{S_o} &= \|S_{o_h} - S_o\|_{L^2(\Omega,T)}, & E_{p_w} &= \|p_{w_h} - p_w\|_{L^2(\Omega,T)}, \\ E_{\mathbf{q}_S} &= \|\mathbf{q}_{S_h} - \mathbf{q}_S\|_{L^2(\Omega,T)}, & E_{\mathbf{q}_p} &= \|\mathbf{q}_{p_h} - \mathbf{q}_p\|_{L^2(\Omega,T)}, \\ E_{S^*_o} &= \|S^*_{o_h} - S^*_o\|_{L^2(\Omega,T)}, & E_{p^*_w} &= \|p^*_{w_h} - p^*_w\|_{L^2(\Omega,T)}. \end{aligned} \tag{39}$$

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

$$\begin{aligned} S_o &= \sin(\pi x)\sin(\pi y)\sin(t), \\ p_w &= \cos(\pi x)\cos(\pi y)\cos(t), \end{aligned} \tag{40}$$

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 $\cdot$ s and the water viscosity as $\mu_w = 0.1$ Pa $\cdot$ s. Specifically for this example, we use the Brooks-Corey model in Equation (1) with $p_e = 0.5$ Pa and $\theta = 1$. We select $\tau_p = 10$ and $\tau_s = 10$, see Equation (9). 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 (40). 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 $h_e/\Delta t$.

Figure 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 (30) and (31), 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, $p^*_{w_h}$.

9.2 | Artificial viscosity analysis

In this example, we analyze the behaviour 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 2. Therefore, water will mobilize the oil from left to right and the flux will be parallel to the no-flow boundaries, Γ_{nf}.

The prescribed boundary conditions are:

$$\begin{aligned} p_w^{\Gamma_{in}} &= 3 \cdot 10^6 \text{ Pa}, & S_o^{\Gamma_{in}} &= 0.3, \text{ on } \Gamma_{in}, \\ p_w^{\Gamma_{out}} &= 10^6 \text{ Pa}, & \left(\frac{\lambda_o \lambda_w}{\lambda_t}\mathbf{K}\nabla p_c\right)\cdot\mathbf{n} &= 0, \quad \text{on } \Gamma_{out}, \\ \mathbf{v}_t \cdot \mathbf{n} &= 0, & \mathbf{v}_o \cdot \mathbf{n} &= 0, \quad \text{on } \Gamma_{n_f}, \\ f_w &= 0, & f_o &= 0, \quad \text{in } \Omega. \end{aligned} \tag{41}$$

The parameters of the Brooks-Corey model are $p_e = 10^3$ Pa and $\theta = 2$, and the residuals saturations 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 (33).

Figure 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 3(a) and 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 (34) and (35). As a consequence of this

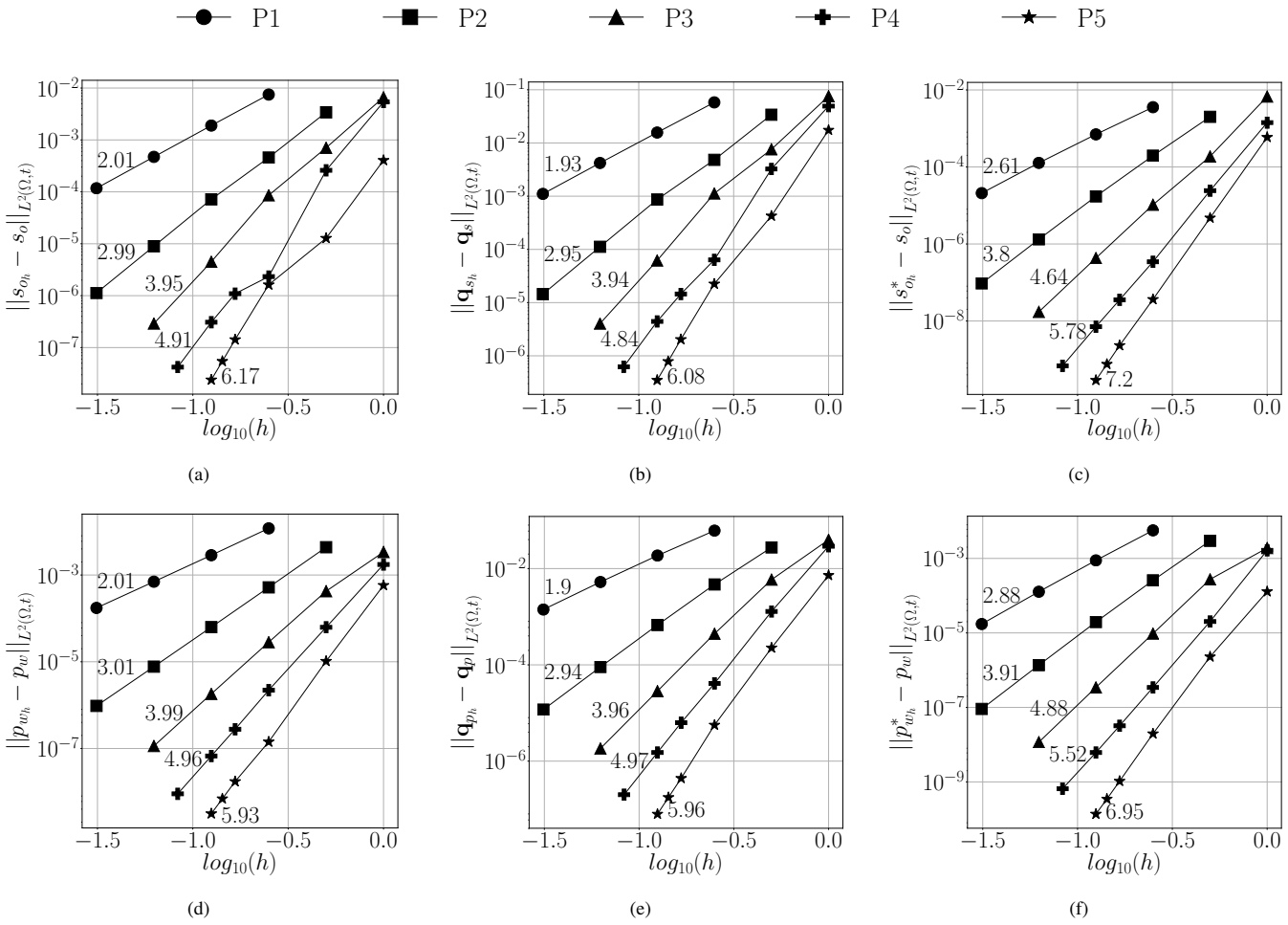

FIGURE 1 Convergence rates for: a) oil saturation, b) oil saturation flux, c) post-processed oil saturation, d) water pressure; e) water pressure flux, and f) post-processed water pressure.

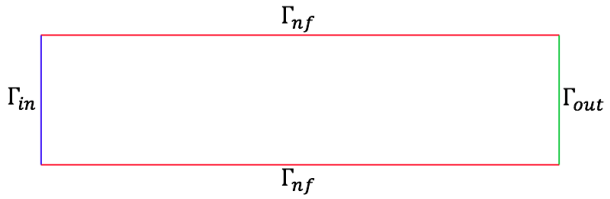

FIGURE 2 Rectangular domain and associated boundary conditions.

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.

9.3 | 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 9.2. 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 II5 and GL6. These are time integration schemes of orders from one to six. For all of them,

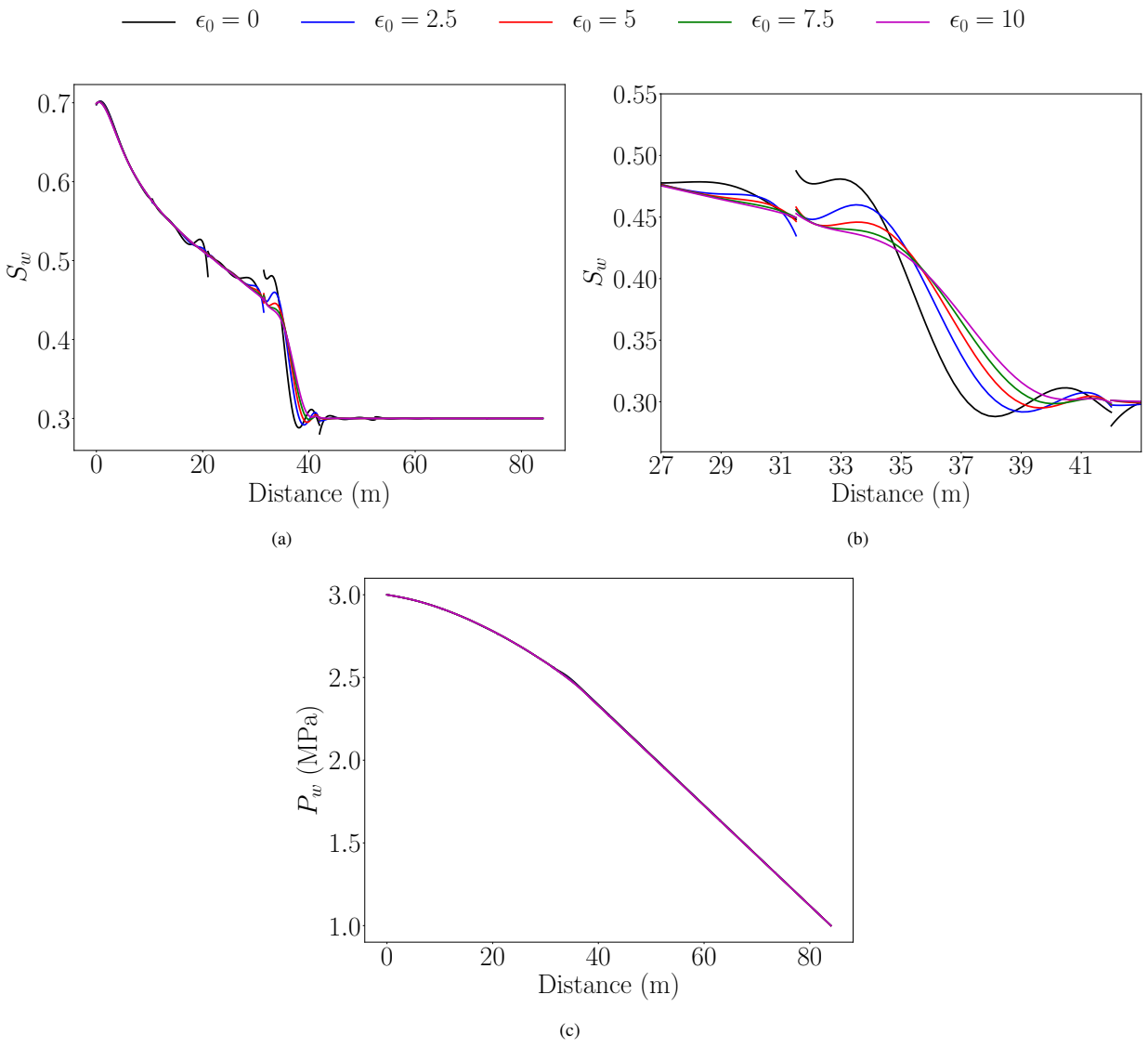

FIGURE 3 Plot over the line $y = 0$ at 22 days using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with $h_e = 10.5$ meters, and using GL8 scheme with $\Delta t = 2$ days for different values of ϵ_0: a) water saturation, b) zoom around the sharp front of the water saturation, and c) water pressure.

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 (33).

Figure 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 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.

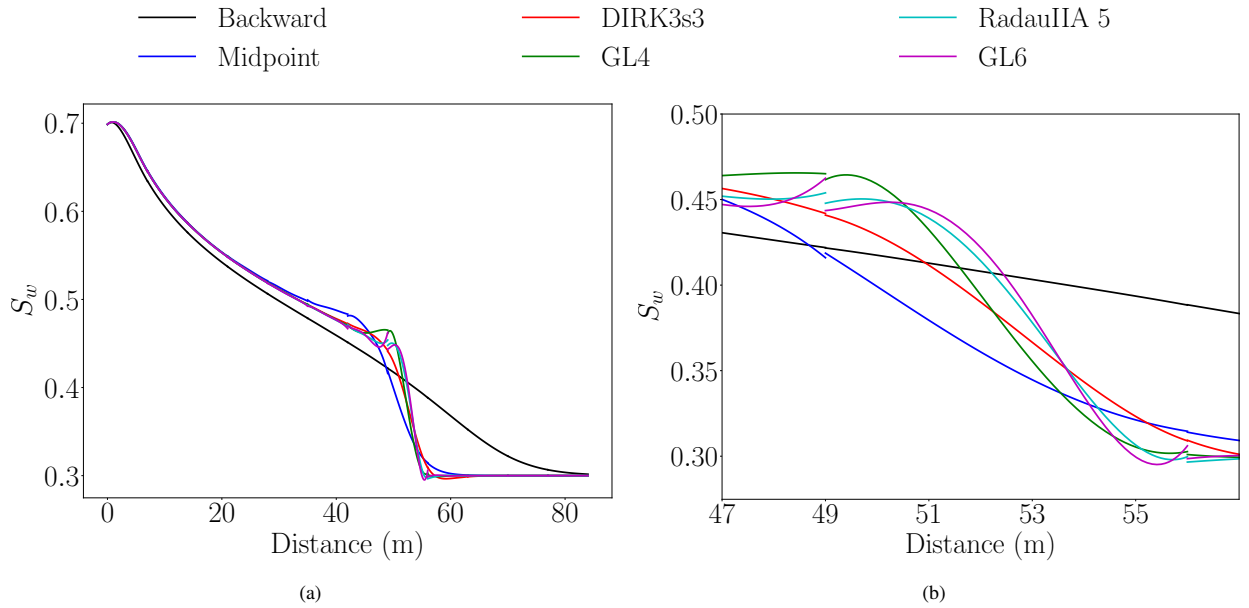

FIGURE 4 Plot over the line $y = 0$ at 30 days using a mesh composed of quadrilateral elements of polynomial degree $P = 4$ with size $h_e = 7$ meters, and using several time integration schemes with $\Delta t = 2$ days: a) water saturation profile, and b) zoom around the sharp front.

9.4 | 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 9.2.

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 $P = 6$ with size $h_1 = 5.25$ meters, and GL8 scheme with $\Delta t = 1$ day. 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 $P = 2$ with size $h_3 = 1.75$ meters, and GL4 scheme with $\Delta t = 0.5$ days. We compute the amount of artificial viscosity using $\kappa = 6$, $s_0 = -10$ and $\epsilon_0 = 2$ in Equation (33). The amount of artificial viscosity is the same in all the cases since the resolution of the spatial discretization is the same.

Figure 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.

9.5 | 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 9.2. 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 (33).

Figure 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

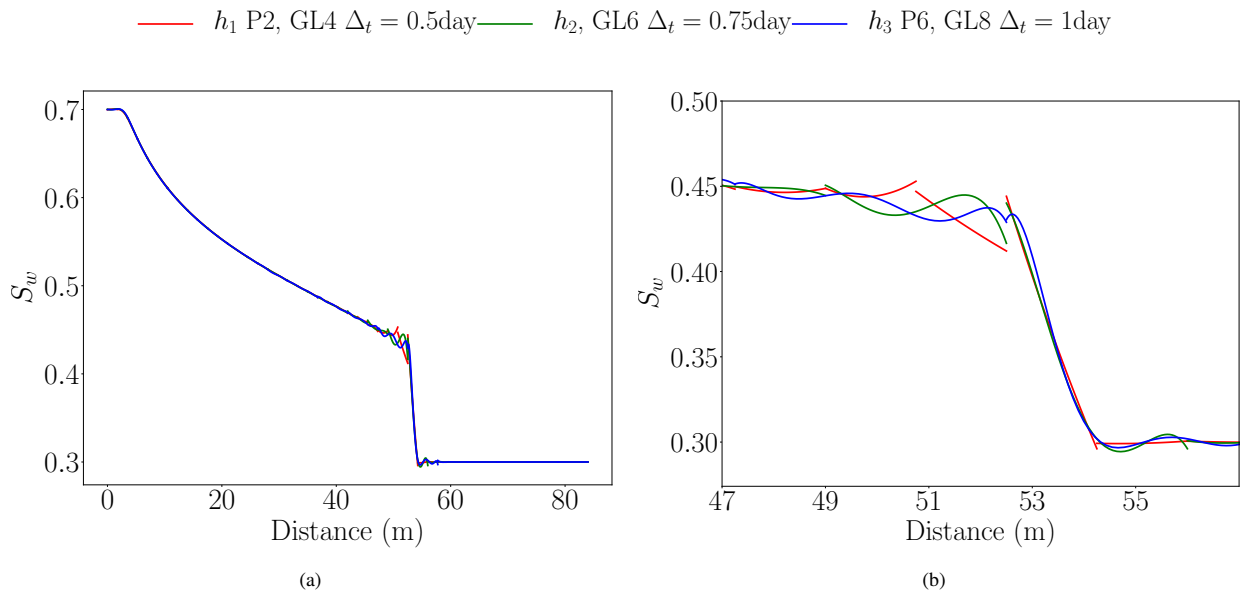

FIGURE 5 Plot over the line $y = 0$ at 30 days using three different spatial and temporal discretizations with the same resolution: a) water saturation profile and b) zoom around the sharp front of the water saturation profile.

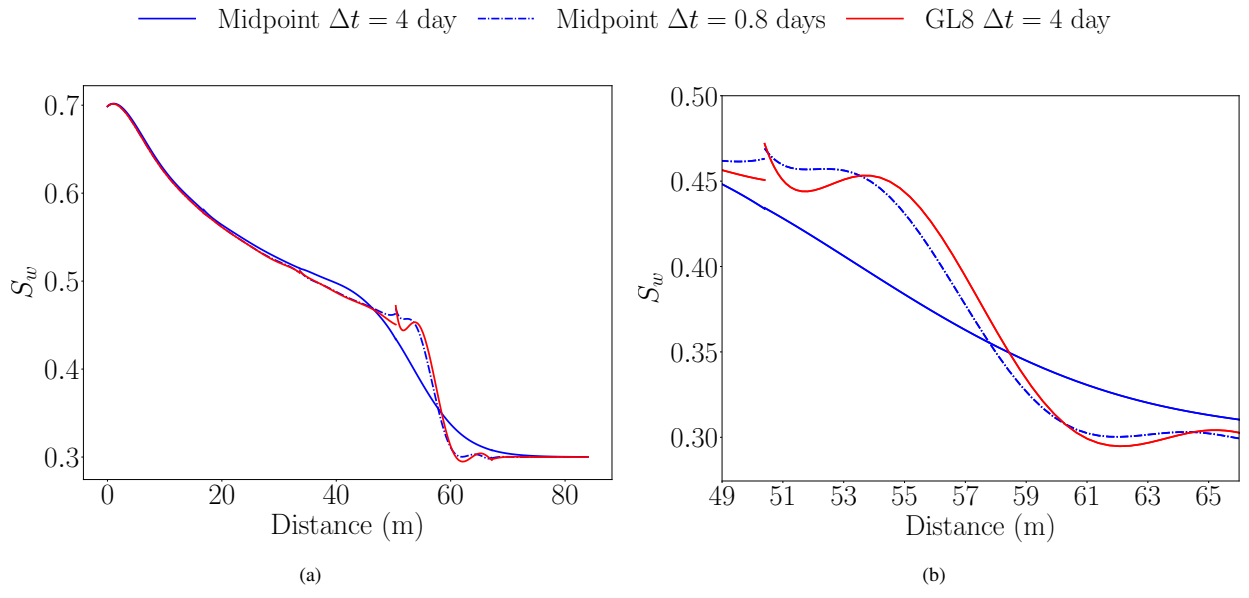

FIGURE 6 Plot over the line $y = 0$ at 32 days using a mesh composed of quadrilateral elements of polynomial degree $P = 6$ with $h_e = 16.8$ meters, and using the midpoint scheme with $\Delta t = 4$ and $\Delta t = 0.8$ days and GL8 scheme with $\Delta t = 4$: a) water saturation profile, and b) zoom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

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.

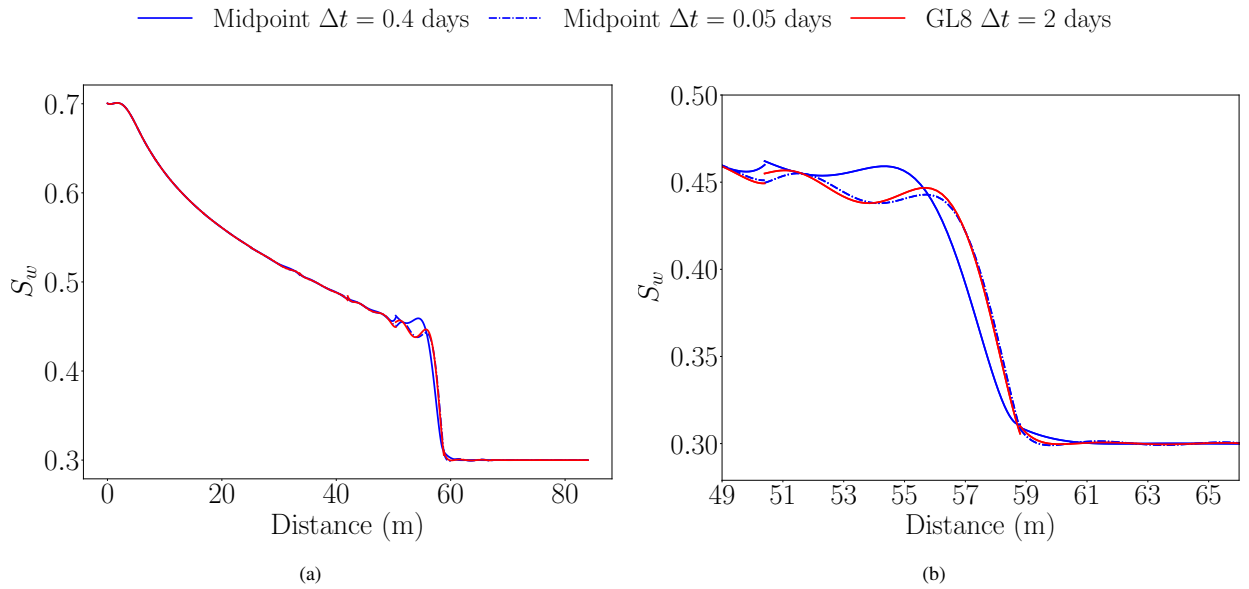

FIGURE 7 Plot over the line $y = 0$ at 32 days using a mesh composed of quadrilateral elements of polynomial degreee $P = 6$ with $h_e = 8.4$ meters, and using the midpoint scheme with $\Delta t = 0.4$ and $\Delta t = 0.05$ days and GL8 scheme with $\Delta t = 2$ days: a) water saturation profile, and b) zoom around the sharp front.

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 (33).

Figure 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 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.

9.6 | Waterflooding in an heterogeneous material with obstacles

In this example, we simulate a case of waterflooding technique through a domain with two different material, see Figure 8. 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 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 9.2.

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.

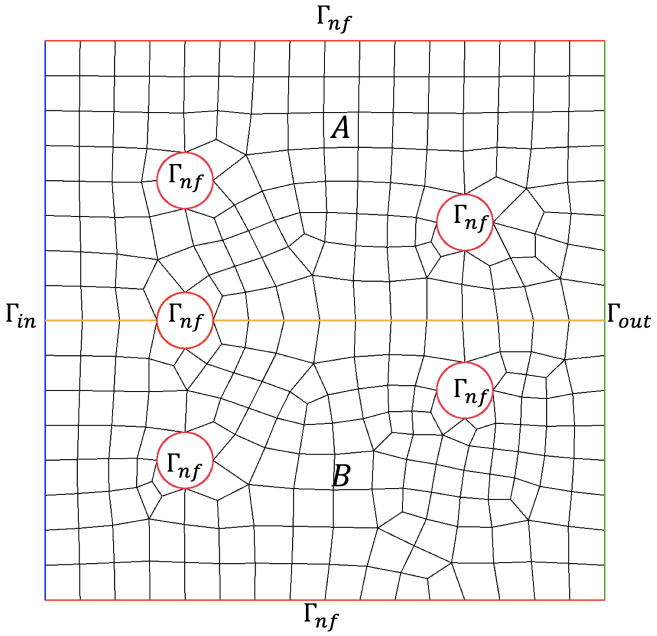

FIGURE 8 Materials, mesh and boundary conditions for the simulation of waterflooding through heterogeneous domain with obstacles.

We discretize the domain using 283 unstructured quadrilateral elements of polynomial degree four (4692 nodes), see Figure 8. 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 (33).

Figure 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 artificial viscosity allows performing the simulation since non-physical saturation values are not obtained, and allows obtaining a solution without oscillations.

Figure 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 11 plots the magnitude of Darcy's water and oil velocities at times $t = 15$ days and $t = 35$ days. Note that Darcy's 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's 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.

9.7 | Nine-spot pattern

In this example, we perform a waterflooding simulation for a nine spot pattern. We consider 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 is considered as no-flow, Γ_{nf}, see Figure 12. The radius of the wells is $r_w = 5$ m. We prescribe the same boundary conditions and Brooks-Corey parameters as in Example 9.2.

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, respectively.

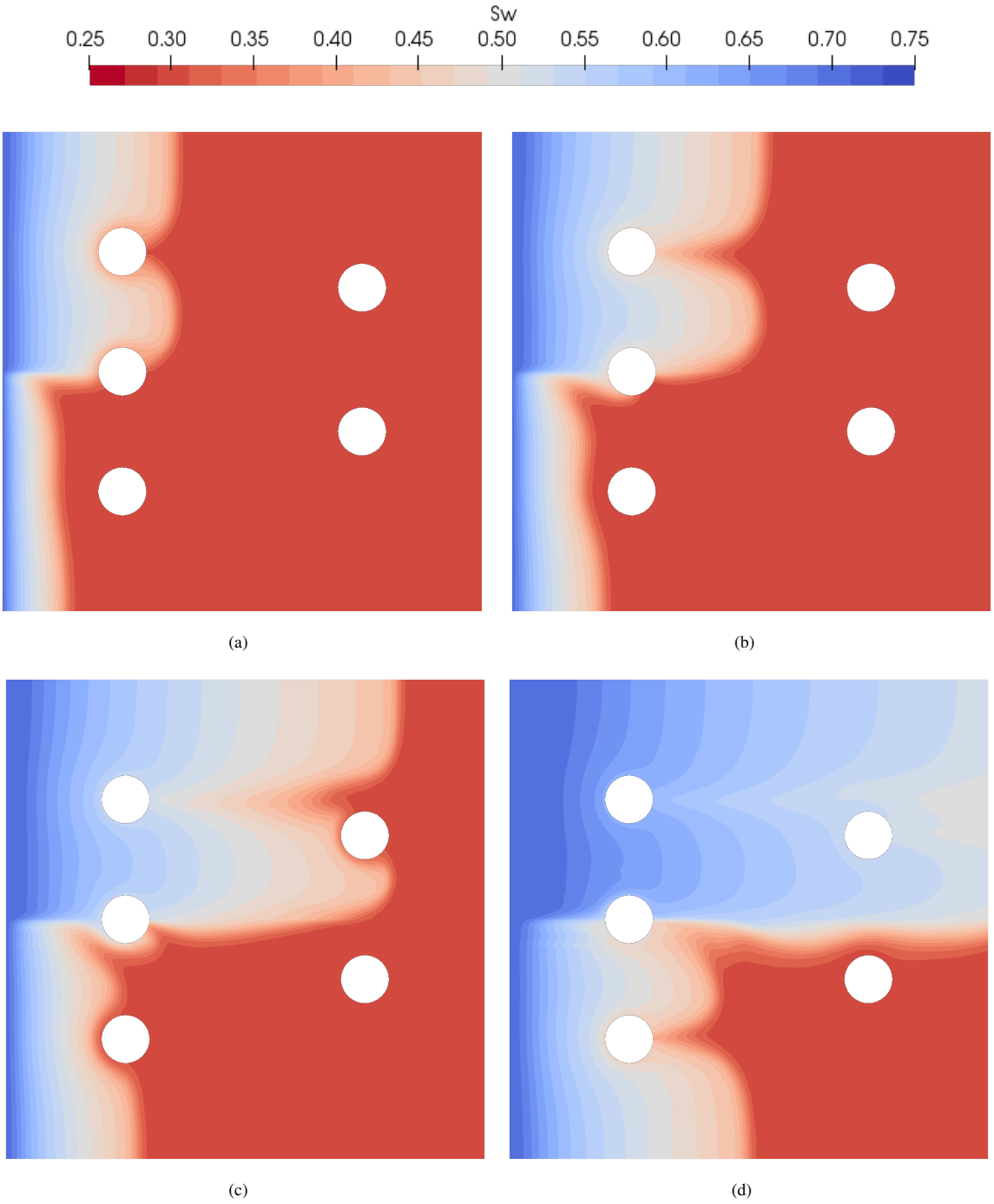

FIGURE 9 Water saturation approximation at time: a) 11 days, b) 15 days, c) 21 days, and d) 35 days.

We discretize the domain with 1114 non-constant size unstructured quadrilateral elements of polynomial degree four (18200 nodes), see Figure 12. 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 (33), being $h_e = \sqrt{\int_e 1 \mathrm{d}\Omega}$

Figure 13 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.

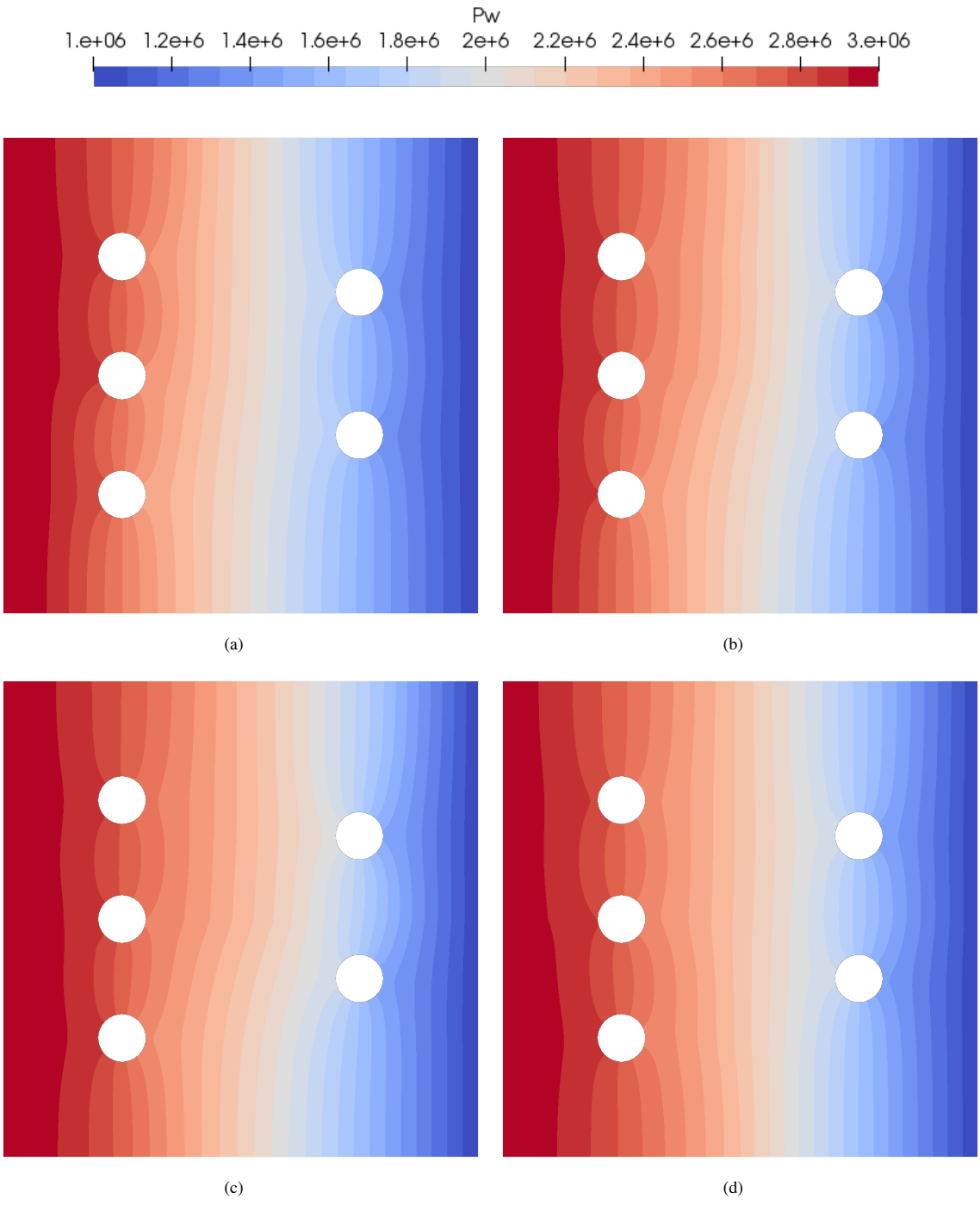

FIGURE 10 Water pressure approximation at time: a) 11 days, b) 15 days, c) 21 days, and d) 35 days.

Figure 14 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 15 plots the magnitude of Darcy's water and oil velocities at times 7.5 and 17.25 days. We observe that the water velocity is higher around the injector wells, whereas the oil velocity is higher around the extractor well. When the waterfront reaches the extractor well, both phases, oil and water, are extracted. From now on, the water velocity increases while the oil velocity decreases around the extractor well.

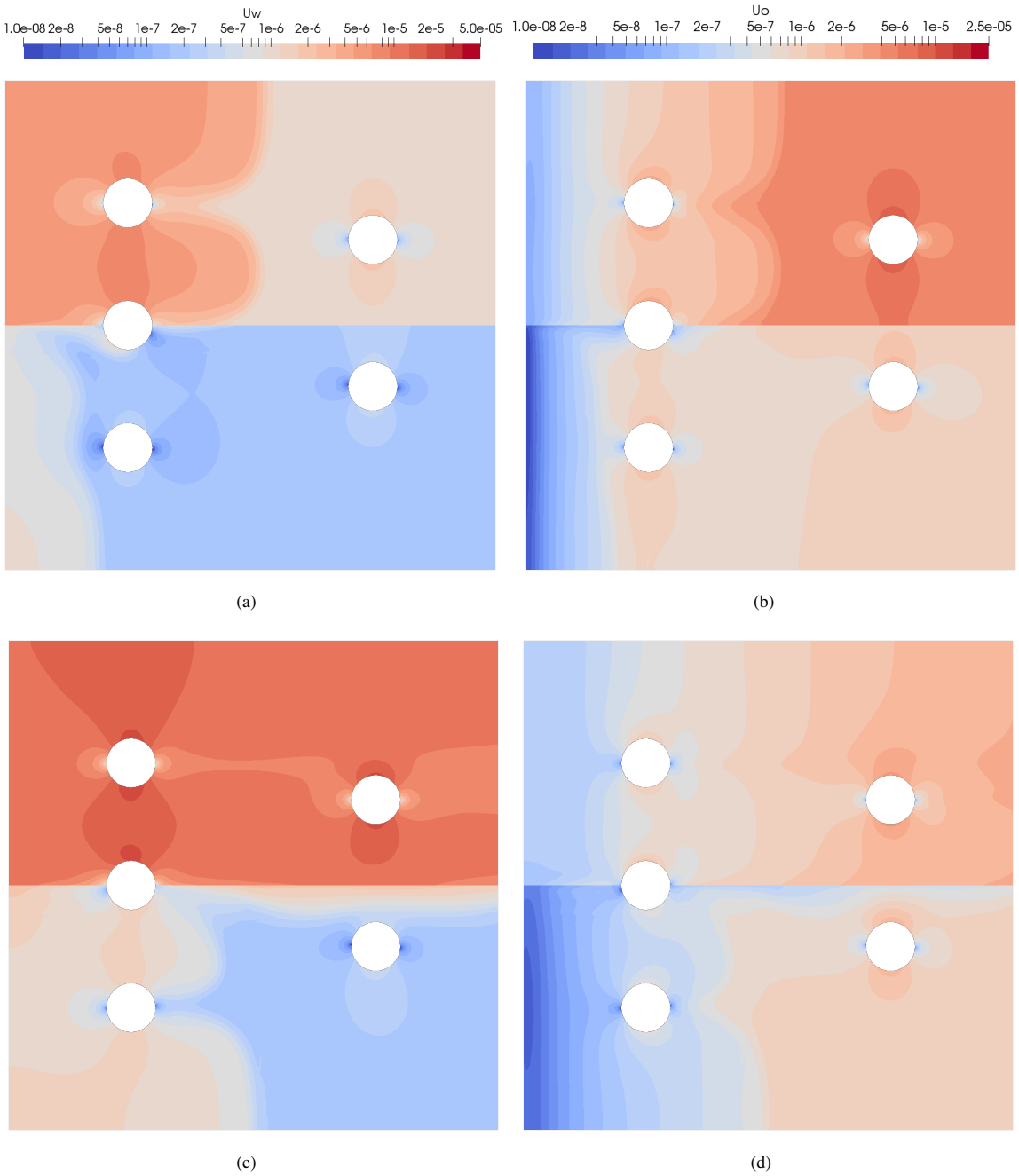

FIGURE 11 Magnitude of the Darcy's velocities for the: a) water phase at time 15 days, b) oil phase at time 15 days, c) water phase at time 35 days and d) oil phase at time 35 days.

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.

10 | CONCLUSIONS AND FUTURE WORK

In this work, we present a memory-efficient high-order hybridizable discontinuous Galerkin formulation combined with high-order fully implicit time integration schemes for two-phase flow through porous media problem. We assume that the fluids are

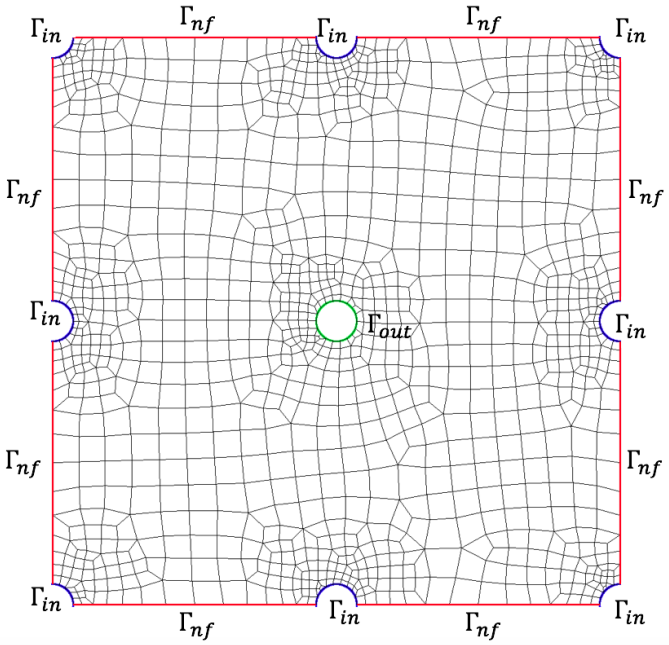

FIGURE 12 Mesh and boundary conditions distributions for the nine-spot pattern example

immiscible and the fluids and the porous media are incompressible. We set the water pressure and oil saturation as the main unknowns. We propose 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 oil saturation equation using the Newton-Raphson method. 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 also propose a temporal integration scheme in terms of the oil saturation instead of its temporal derivative. In this way, we obtain a better sparsity pattern of the Jacobian with less coupling between the stages[27]. Finally, to deal with the sharp fronts that can appear in the saturation profile, we introduce an artificial viscosity term in the saturation equation. To detect the location of the sharp fronts, we use the shock sensor introduced in[31], 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 propose 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 present 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 without 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 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 interacts between them, heterogeneous material properties and unstructured high-order curved meshes.

In the proposed method, we use a Lagrangian basis of shape functions to define the elemental polynomial spaces and therefore, the unknowns of the problem are the nodal values. Other bases of the polynomial space could be used, such as orthonormal polynomials. Thus, the unknowns of the problem would be the coefficients of the polynomial expansion of the solution in

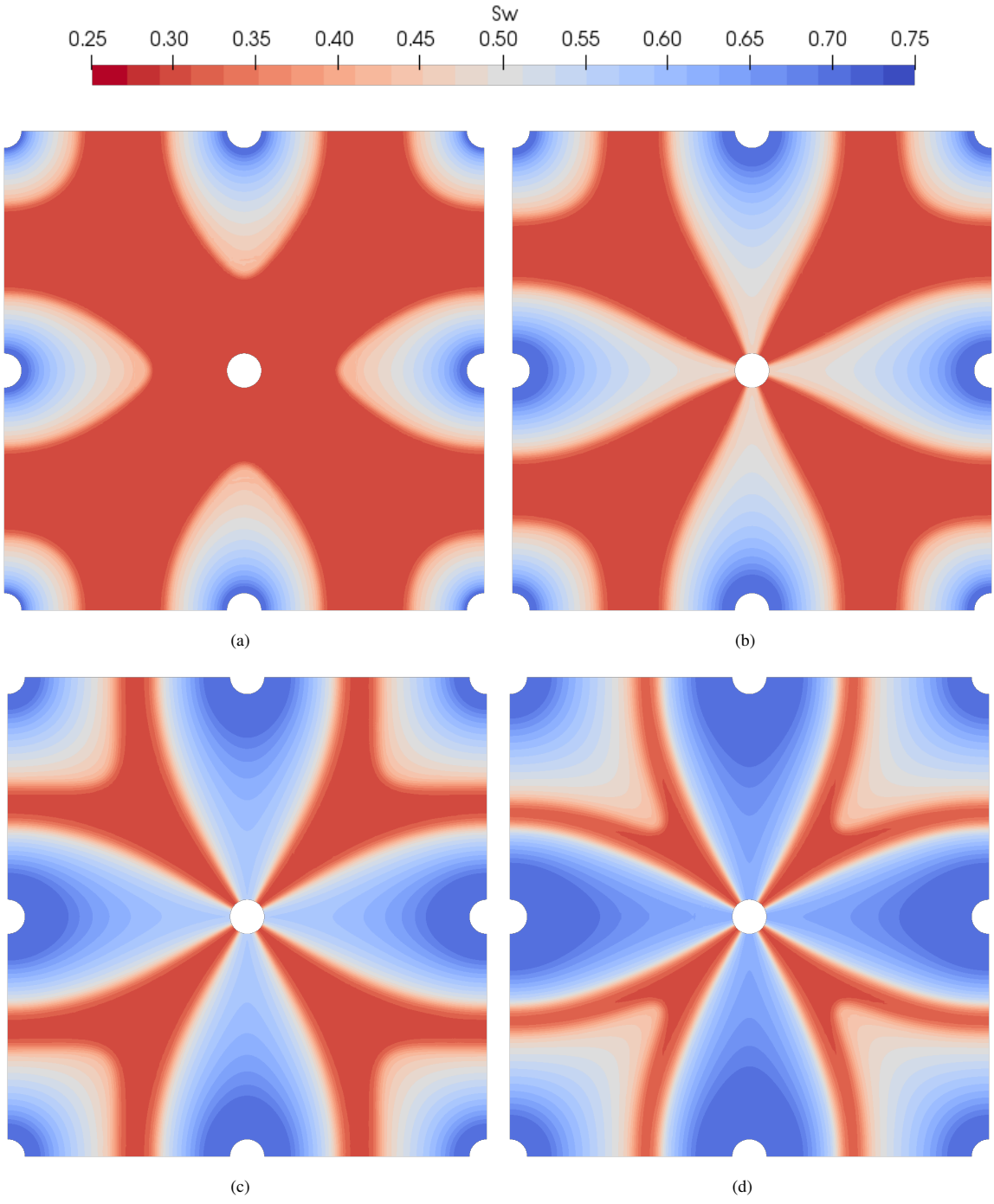

FIGURE 13 Water saturation approximation at time: a) 5 days, b) 7.5 days, c) 12.5 days and d) 17.25 days.

the used basis. Nevertheless, in our work, we selected the Lagrangian basis, since we use a non-uniform nodal distribution that approximately minimizes the Lebesgue constant[48]. The used basis is well-suited for high-order Lagrange interpolation. Moreover, the solution to the problem is the nodal values, which is convenient for practical purposes.

Several aspects can be implemented and analyzed in the future. First, we are interested in increasing the computational efficiency of the formulation. 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

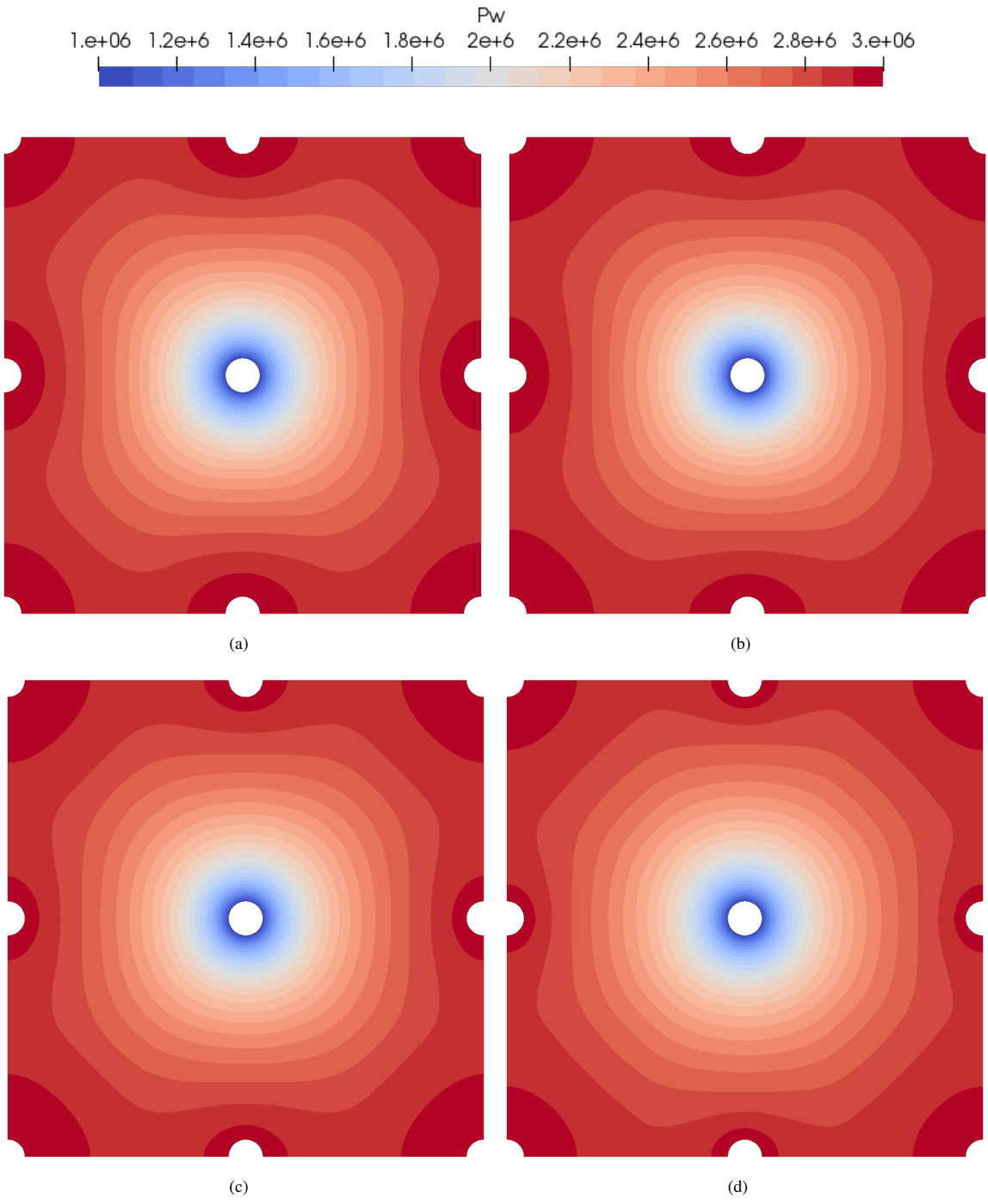

FIGURE 14 Water pressure approximation at time: a) 5 days,b) 7.5 days, c) 12.5 days and d) 17.25 days.

solvers. Thus, we need to investigate efficient iterative linear solvers for fully implicit time integration schemes and appropriated pre-conditioners. Second, we propose to use h-p adaptivity to increase the accuracy while reducing the oscillations and computational cost. That is, we will be able to use small low-order elements near the front and large high-order elements for the rest of the domain. Third, we will analyze the possible relationship between the amount of artificial viscosity and the time integration scheme. That is, low-order integration schemes introduce temporal dissipation and therefore, we may need to introduce less artificial viscosity. Fourth, we will further investigate the stability of the HDG formulation for two-phase flow and select

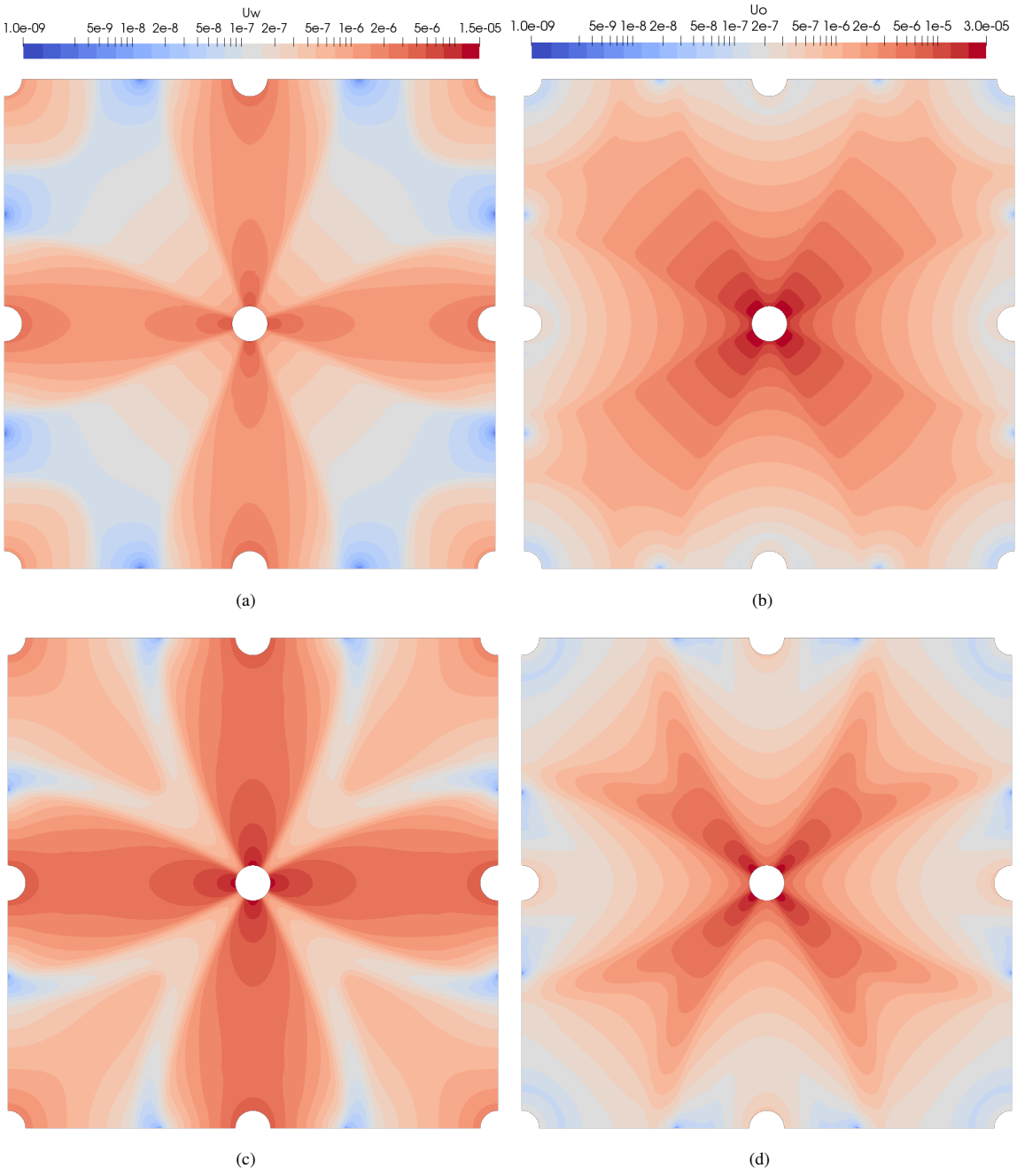

FIGURE 15 Magnitude of Darcy's velocities for the: a) water phase at time 7.5 days,b) oil phase at time 7.5 days, c) water phase at time 17.25 days and d) oil phase at time 17.25 days.

appropriate values of the stability parameter, τ. Finally, we will extend the formulation to other models, such as the black oil model and the compositional flow. Therefore, we will be able to simulate more realistic and complex scenarios.

ACKNOWLEDGEMENTS

This work has been supported by FEDER and the Spanish Government, Ministerio de Economía y Competitividad grant project contract CTM2014-55014-C3-3-R, Ministerio de Ciencia Innovación y Universidades grant project contract PGC2018-097257-B-C33 and the grant BES-2015-072833.

11 | DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request

References

1. Donaldson EC, Chilingarian GV, Yen TF. *Enhanced oil recovery, I: fundamentals and analyses*. Elsevier . 1985.
2. Chen Z, Huan G, Ma Y. *Computational methods for multiphase flows in porous media*. SIAM . 2006.
3. Radu FA, Nordbotten JM, Pop LS, Kumar K. A robust linearization scheme for finite volume based discretizations for simulation of two-phase flow in porous media. *J Comput Appl Math* 2015; 289: 134–141.
4. Salinas P, Pavlidis D, Xie Z, Osman H, Pain CC, Jackson MD. A discontinuous control volume finite element method for multi-phase flow in heterogeneous porous media. *J Comput Phys* 2018; 352: 602–614.
5. Salinas P, Pavlidis D, Xie Z, et al. Improving the robustness of the control volume finite element method with application to multiphase porous media flow. *Int J Numer Methods Fluids* 2017; 85(4): 235–246.
6. Hughes TJ, Masud A, Wan J. A stabilized mixed discontinuous Galerkin method for Darcy flow. *Comput Methods Appl Mech Eng* 2006; 195(25-28): 3347–3381.
7. Hou J, Chen J, Sun S, Chen Z. Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media. *J Comput Appl Math* 2016; 307: 262–283.
8. Rivière B, Wheeler MF. Discontinuous Galerkin methods for flow and transport problems in porous media. *Commun Numer Methods Eng* 2002; 18(1): 63–68.
9. Rivière B. Analysis of a discontinuous finite element method for the coupled Stokes and Darcy problems. *J Sci Comput* 2005; 22(1): 479–500.
10. Klieber W, Rivière B. Adaptive simulations of two-phase flow by discontinuous Galerkin methods. *Comput Methods Appl Mech Eng* 2006; 196(1-3): 404–419.
11. Epshteyn Y. *HP primal discontinuous Galerkin finite element methods for two-phase flow in porous media*. PhD thesis. University of Pittsburgh, 2007.
12. Ern A, Mozolevski I, Schuh L. Discontinuous Galerkin approximation of two-phase flows in heterogeneous porous media with discontinuous capillary pressures. *Comput Methods Appl Mech Eng* 2010; 199(23-24): 1491–1501.
13. Badia S, Codina R. Stabilized continuous and discontinuous Galerkin techniques for Darcy flow. *Comput Methods Appl Mech Eng* 2010; 199(25-28): 1654–1667.
14. Arbogast T, Juntunen M, Pool J, Wheeler MF. A discontinuous Galerkin method for two-phase flow in a porous medium enforcing H (div) velocity and continuous capillary pressure. *Comput Geosci* 2013; 17(6): 1055–1078.
15. Li J, Rivière B. High order discontinuous Galerkin method for simulating miscible flooding in porous media. *Comput Geosci* 2015; 19(6): 1251–1268.

16. Bastian P. A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure. *Comput Geosci* 2014; 18(5): 779–796.

17. Jamei M, Ghafouri H. A novel discontinuous Galerkin model for two-phase flow in porous media using an improved IMPES method. *Int J Numer Methods Heat Fluid Flow* 2016; 26(1): 284–306.

18. Fabien MS, Knepley M, Rivière B. A hybridizable discontinuous Galerkin method for two-phase flow in heterogeneous porous media. *Int J Numer Methods Eng* 2018; 116(3): 161–177.

19. Fabien MS, Knepley M, Rivière B. A high order hybridizable discontinuous Galerkin method for incompressible miscible displacement in heterogeneous media. *RINAM* 2020: 100089.

20. Costa-Solé A, Ruiz-Gironés E, Sarrate J. An HDG formulation for incompressible and immiscible two-phase porous media flow problems. *Int J Comput Fluid D* 2019; 33(4): 137–148.

21. Babuska I, Szabo BA, Katz IN. The p-version of the finite element method. *SIAM J Numeric Anal* 1981; 18(3): 515–545.

22. Löhner R. Error and work estimates for high-order elements. *Int J Numer Methods Fluids* 2011; 67(12): 2184–2188.

23. Wang Z, Fidkowski R, Bassi F, et al. High-order CFD methods: current status and perspective. *Int J Numer Methods Fluids* 2013; 72(8): 811–845.

24. Löhner R. Improved error and work estimates for high-order elements. *Int J Numer Methods Fluids* 2013; 72(11): 1207–1218.

25. Huerta A, Angeloski A, Roca X, Peraire J. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int J Numer Methods Eng* 2013; 96(9): 529–560.

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

27. Pazner W, Persson PO. Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations. *J Comput Phys* 2017; 335: 700–717.

28. Montlaur A, Fernández-Méndez S, Huerta A. High-order implicit time integration for unsteady incompressible flows. *Int J Numer Methods Fluids* 2012; 70(5): 603–626.

29. Butcher JC. *Numerical methods for ordinary differential equations*. John Wiley & Sons . 2016.

30. Fernández P, Christophe A, Terrana S, Nguyen NC, Peraire J. Hybridized discontinuous Galerkin methods for wave propagation. *J Sci Comput* 2018; 77(3): 1566–1604.

31. Persson PO, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. *44th AIAA Aerospace Sciences Meeting and Exhibit* 2006: 112.

32. Huerta A, Casoni E, Peraire J. A simple shock-capturing technique for high-order discontinuous Galerkin methods. *Int J Numer Methods Fluids* 2012; 69(10): 1614–1632.

33. Casoni E, Peraire J, Huerta A. One-dimensional shock-capturing for high-order discontinuous Galerkin methods. *Int J Numer Methods Fluids* 2013; 71(6): 737–755.

34. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. *J Comput Phys* 2009; 228(9): 3232–3254.

35. Nguyen NC, Peraire J, Cockburn B. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. *J Comput Phys* 2009; 228(23): 8841–8855.

36. Nguyen NC, Peraire J, Cockburn B. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *J Comput Phys* 2011; 230(10): 3695–3718.

37. Kirby RM, Sherwin SJ, Cockburn B. To CG or to HDG: a comparative study. *J Sci Comput* 2012; 51(1): 183–212.

38. Sevilla R, Huerta A. Tutorial on Hybridizable Discontinuous Galerkin (HDG) for second-order elliptic problems. In: Advanced Finite Element Technologies. Springer. 2016 (pp. 105–129).

39. Butcher JC. On Runge-Kutta processes of high order. *J Aust Math Soc* 1964; 4(2): 179–194.

40. Russell TF, Wheeler MF. Finite element and finite difference methods for continuous flows in porous media. In: The mathematics of reservoir simulation. SIAM. 1983 (pp. 35–106).

41. Zhu J, Taylor Z, Zienkiewicz O. *The finite element method: its basis and fundamentals*. Elsevier . 2013.

42. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *J Comput Phys* 1997; 131(2): 267–279.

43. Butcher JC. Implicit runge-kutta processes. *Math Comput* 1964; 18(85): 50–64.

44. Butcher JC. A history of Runge-Kutta methods. *Appl Numer Math* 1996; 20(3): 247–260.

45. Von Neumann J, Richtmyer RD. A method for the numerical calculation of hydrodynamic shocks. *J Appl Phys* 1950; 21(3): 232–237.

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

47. Corey AT. Hydraulic properties of porous media. *Colorado State University, Hydraulic Papers* 1964(3).

48. Warburton T. An explicit construction of interpolation nodes on the simplex. *J Eng Math* 2006; 56(3): 247–262.