Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

High-order hybridizable discontinuous Galerkin formulation for one-phase flow through porous media

Albert Costa-Solé · Eloi Ruiz-Gironés · Josep Sarrate.

Received: date / Accepted: date

Abstract We present a stable high-order hybridizable discontinuous Galerkin (HDG) formulation coupled with high-order diagonal implicit Runge-Kuta (DIRK) schemes to simulate slightly compressible one-phase flow through porous media. The HDG stability depends on the selection of a single parameter and its definition is crucial to ensure the stability and to achieve the high-order properties of the method. Thus, we extend the work of Nguyen et al. in J. Comput. Phys. 228:8841–8855, 2009 to deduce an analytical expression for the stabilization parameter using the material parameters of the problem and the Engquist-Osher monotone flux scheme. The formulation is high-order accurate for the pressure, the flux and the velocity with the same convergence rate of P+1, being P the polynomial degree of the approximation. This is important because high-order methods have the potential to reduce the computational cost while obtaining more accurate solutions with less dissipation and dispersion errors than low order methods. The formulation can use unstructured meshes to capture the heterogeneous properties of the reservoir. In addition, it is conservative at the element level, which is important when solving PDE's in conservative form. Moreover, a hybridization procedure can

Albert Costa-Solé
Laboratori de Càlcul Numèric (LaCàN),Departament d'Enginyeria Civil i Ambiental, ETSECCPB, Universitat Politècnica de Catalunya - BarcelonaTech, Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: albert.costa@upc.edu

Eloi Ruiz-Gironés
Barcelona Supercomputing Center - BSC, www.bsc.es, Jordi Girona 31, 08034 Barcelona, Spain
E-mail: eloi.ruizgirones@bsc.es

Josep Sarrate
Laboratori de Càlcul Numèric (LaCàN),Departament d'Enginyeria Civil i Ambiental, ETSECCPB, Universitat Politècnica de Catalunya - BarcelonaTech, Jordi Girona 1-3, 08034 Barcelona, Spain
E-mail: jose.sarrate@upc.edu

This is a post-peer-review, pre-copyedit version of an article published in *Journal of Scientific Computing*. The final authenticated version is available online at:
http://dx.doi.org/10.1007/s10915-021-01436-9

be applied to reduce the size of the global linear system. To keep these advantages, we use DIRK schemes to perform the time integration. DIRK schemes are high-order accurate and have a low memory footprint. We show numerical evidence of the optimal convergence rates obtained with the proposed formulation. Finally, we present several examples to illustrate the capabilities of the formulation.

Keywords One-phase porous media flow · slightly compressible · hybridizable discontinuous Galerkin · high-order · diagonally implicit Runge-Kutta

1 Introduction

One-phase flow through porous media formulation is used to model several engineering problems. In particular, one of the most important aspects in the management, planning and analysis of the environmental impact of oilfields exploitation is the reliable prediction of its behavior under different operating conditions. For instance, using accurate numerical simulations, engineers can evaluate the hydrocarbon flow and pressure drop in order to optimize the hydrocarbon production while minimizing the environmental impact, see [9, 8].

In this paper, we focus on the primary oil recovery stage, which begins when the first well is drilled, and corresponds to 10% of the total oil production [8]. During this stage, the pressure difference between the surface and the reservoir is high enough to move the hydrocarbon upward [46]. The governing equation for this scenario is a non-linear transient partial differential equation (PDE), which is obtained from the combination of the mass conservation with Darcy's law and equations of state for the fluid and the rock [8].

To obtain accurate approximations of the pressure and the Darcy velocity, several requirements have to be fulfilled. The formulation should deal with unstructured meshes to capture heterogeneous and complex subsurface configurations. Moreover, it should provide high-accurate solutions without hampering the computational cost. Therefore, unstructured high-order formulations are well-suited for these applications. In addition, the formulation has to be stable and the mass should be conserved, at least, at the element level. Nowadays, several methods such as the finite differences (FD), finite volumes (FV), continuous finite elements (CG), mixed finite elements (Mixed CG), and discontinuous Galerkin methods (DG), have been applied.

Recently, many efforts have been focused on applying high-order methods to these kind of problems because of their advantages [12,28,14,15]. 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 [4,29,49]. Hence, it has been shown that high-order spatial discretization methods can be more accurate than low-order ones for the same mesh resolution, that is, for the same number of degrees of freedom [49]. Moreover, it is also reported that high-order methods introduce less diffusion and dispersion errors in the solutions [4,29,49]. 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 [29,49,30,20,18].

Nevertheless, to obtain these advantages in unsteady problems, the temporal integration error has to be low enough. In general, there are two ways of controlling the temporal error. The first one is to use low-order temporal schemes with small time steps, and the second one is to use high-order temporal schemes with large time steps. However, if high temporal accuracy is required, low-order time integration schemes may require prohibitively small time steps. Thus, the computational cost of the simulation can be severely hampered because, at each time step, a non-linear problem has to be solved. In these cases, high-order temporal schemes with large time steps may alleviate the computational cost [11,38].

The first contribution of this paper is to develop a hybridizable discontinuous Galerkin (HDG) formulation for the slightly compressible one-phase flow problem. HDG provides built-in stabilization for arbitrary polynomial degrees that depends on a single parameter, τ. The selection of a correct value of τ leads to a stable method that is also consistent and convergent, see [35,36]. HDG has element-wise mass conservation and can handle the heterogeneous reservoir properties and its geometric complexities, since unstructured meshes can be used. The method obtains a convergence rate for the pressure and the flux of order $P+1$ in L^2-norm, when element-wise polynomials of degree $P \geq 0$ are used and the error of the temporal discretization is low enough, [25, 35,36,47]. Consequently, the Darcy velocity also converges with order $P+1$, since it depends on the pressure and its flux. Moreover, an element-wise post-processing can be applied at chosen time steps to obtain a $P+2$ convergence rate for the pressure [35,36,47]. Furthermore, the computational cost of solving the linear system is also reduced since a hybridization procedure is applied. In addition, we couple the high-order spatial HDG discretization with high-order DIRK schemes to obtain highly-accurate solutions in space and time. Thus, we obtain a non-linear system at each stage of the DIRK scheme that we solve sequentially, stage-by-stage, using the Newton-Raphson method.

The second contribution of this paper is the specific choice for the stabilization parameter, τ. This is one of the key aspects of the HDG since it ensures the existence and uniqueness of the obtained approximation, as well as the stability of the formulation, see details in [25]. According to [35,36], we split the stabilization parameter into diffusive and convective parts. The diffusive part has to be positive [35,36,25], and we select it according to the physical parameters of the problem. The convective part is selected using a monotone scheme flux [27,35,36]. Specifically, we use the Engquist-Osher monotone scheme flux, and we deduce an analytical expression of the convective part. Thus, the presented formulation provides a stable method for arbitrary polynomial degrees. Moreover, since we have an analytical expression of the stabilization parameter, we can introduce its derivatives in the Newton-Raphson solver.

Therefore, in this work we propose a stable, convergent and high-order accurate method for both pressure and Darcy's velocity. We highlight that the

selection of the τ stabilization parameter is crucial to obtain these properties of the method. Specifically, the stability of the method ensures the continuity of the obtained solution respect to the problem data (initial condition, boundary condition, source term, porous media parameters,etc). That is, the stability of the method ensures that small perturbations of the problem data because of numerical errors do not lead to unbounded oscillations in the obtained solution. Moreover, the proposed selection of the τ parameter leads to an optimally accurate high-order method. Both the pressure and the Darcy's velocity converge with an optimal convergence rate of $P + 1$. The higher accuracy of the Darcy's velocity is of special importance, since other methods achieve a convergence rate of P.

In this paper we extend the formulation presented in [36] to one-phase flow through porous media. Specifically, in our work, we deal with temporal, diffusion and convective non-linear terms, whereas in [36] only the convective term is non-linear.

In our approach, we apply the following methodology. First, we convert the second order PDE into a system of two first-order PDE's by introducing a new variable, $\mathbf{q}$, which is related to the pressure gradient, ∇p. Second, we obtain the weak form, in which we introduce the new variable $\hat{p}$, which is the trace of the pressure and is defined on the mesh skeleton. After spatial discretization, we convert the PDE's into a system of differential algebraic equations (DAE). Third, we perform the temporal discretization by using a diagonal implicit Runge-Kutta scheme (DIRK). Thus, we obtain a non-linear system at each stage of the DIRK scheme that we solve using Newton-Raphson method.

The outline of this paper is as follows. In Section 2, we describe the related work. In Section 3, we introduce the HDG formulation for the one-phase flow problem. In Section 4, we show several examples to illustrate the advantages of the proposed formulation. Finally in Section 5, we summarize the main contributions of this paper and we describe the issues that will be addressed in the near future.

2 Related work

Several spatial discretization methods have been used in reservoir simulation. For instance, the finite difference method solves the differential equations by approximating the derivatives with incremental ratios. The mass is locally conserved at the discretization points [8,3,43]. Although this technique is fast, it requires complex implementation to deal with the reservoir geometric complexity and to obtain high-order discretizations.

The finite volume method obtains an element-wise approximation of the solution. It can be applied to unstructured polygonal and polyhedral meshes, and therefore it has geometric flexibility. Moreover, the method is locally conservative at the element level, see [48]. It has been successfully applied as a first-order method in reservoir simulations in references [8,39,45,44].

The continuous Galerkin method approximates the solution using continuous element-wise polynomials of arbitrary degree. The method can deal with the reservoir complexity since the mesh can be adapted to the geometry, see [51,43,8]. The number of unknowns of the linear system can be reduced by applying an hybridization procedure, and therefore the computational cost is reduced. This method is conservative at the domain level. The convergence rate in L^2-norm for the pressure is of order $P+1$, whereas for the flux is of order P. Therefore, the Darcy velocity converges with order P, being that the main drawback of the method for these type of simulations.

The mixed finite element method introduces a new unknown, $\mathbf{q}$, which is related to the pressure gradient, see [8,1]. It is not possible to use arbitrary spaces to approximate p and $\mathbf{q}$ in order to obtain a stable, consistent and convergent method. Nevertheless, references [32,6,21,31] present different stabilization techniques for specific selection of approximation spaces. Since we introduce a new unknown, mixed methods have more unknowns than standard finite element methods, although a hybridization technique can also be applied to reduce the number of unknowns of the linear system.

Discontinuous Galerkin methods have been shown as competitive discretization methods in these type of problems [26,12,28,40,22,19,34,5,2,13]. These methods approximate the pressure using element-wise discontinuous polynomials. They introduce a new discontinuous variable called the flux, $\mathbf{q}$, which is also related to the pressure gradient. Several advantages arise from using DG methods. For instance, mass conservation is verified at the element level, and they can use polynomials of arbitrary degree, which improve the accuracy of the computed approximations, see [25]. Moreover, there are DG formulations that allow solving only for the main unknown, p, to increase the computation efficiency, which is known as primal formulation, see [12].

The hybridizable discontinuous Galerkin method approximates the pressure and the flux, $\mathbf{q}$, using element-wise discontinuous polynomials as DG methods do, see [25,35,36,37]. To relate the unknowns of adjacent elements, this formulation imposes the continuity of the normal component of a numerical flux between adjacent elements. To perform this, the trace of p, usually denoted as $\hat{p}$, is introduced on the mesh skeleton as a new variable. The HDG method provides built-in stabilization for arbitrary polynomial degree and ensures element-wise mass balance. In addition, HDG uses polynomials of arbitrary degree, and it obtains a convergence rate of $P+1$ in the L^2-norm for p and $\mathbf{q}$, when element-wise polynomials of degree $P \geq 0$ provided that the temporal error is low enough, see [25,35,36,47]. Therefore, the Darcy velocity also converges with order $P+1$. Moreover, doing an element-wise post-processing, an improved pressure approximation is computed that converges with a rate of $P+2$ in the L^2-norm. Finally, the method can be always hybridized in terms of $\hat{p}$, reducing the size of the linear system that has to be solved. For these reasons, HDG methodology has been applied in porous media flow problems [14,15,10].

3 Numerical model

3.1 Problem statement

We consider a single Newtonian fluid flow under isothermal conditions that occupies the total soil porosity. We assume that the fluid mass cannot cross the solid interface and mass fluxes can be neglected because dispersion and diffusion are small [8]. Under these assumptions, the governing equations are provided by the mass conservation and the Darcy's law

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = f, \tag{1a}$$

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

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

Following [8], we consider that the fluid and rock compressibility, (c_f, c_r, respectively) are constant in the pressure ranges of the simulation. Moreover, we also assume slightly compressible fluid and rock. Thus, the density and the porosity are approximated as

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

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

$$\phi\rho c_t\frac{\partial p}{\partial t} - \nabla\cdot\left(\frac{\rho}{\mu}\mathbf{K}(\nabla p - \rho\mathbf{g})\right) = f, \tag{3}$$

where $c_t = c_f + c_r$ is the total compressibility, see details in [8].

We define $\Omega \subset \mathbb{R}^d$ as a porous medium with boundary Γ such that $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$, where Γ_D is the Dirichlet boundary and Γ_N is the Neumann boundary. We also consider the time interval $T = [0, t_{end}]$. Therefore, our numerical model for a slightly compressible one-phase flow through porous media is composed of Equation (3) and the corresponding boundary and initial conditions

$$\begin{cases} s(p)\dfrac{\partial p}{\partial t} + \nabla\cdot(-\mathbf{A}(p)\nabla p + \mathbf{F}(p)) = f(\mathbf{x},t) & \forall\mathbf{x}\in\ \Omega, \quad \forall t\in T, \\ p(\mathbf{x},t) = g_D(\mathbf{x},t) & \forall\mathbf{x}\in\ \Gamma_D, \quad \forall t\in T, \\ (-\mathbf{A}(p)\nabla p + \mathbf{F}(p))\cdot\mathbf{n} = g_N(\mathbf{x},t) & \forall\mathbf{x}\in\ \Gamma_N, \quad \forall t\in T, \\ p(\mathbf{x},0) = p^0(\mathbf{x}) & \forall\mathbf{x}\in\ \Omega, \end{cases} \tag{4}$$

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

$$\begin{aligned} s(p) &= \phi(p)\rho(p)c_t, \\ \mathbf{A}(p) &= \frac{\rho(p)}{\mu}\mathbf{K}, \\ \mathbf{F}(p) &= \frac{\rho(p)^2}{\mu}\mathbf{K}\mathbf{g}. \end{aligned} \tag{5}$$

In order to introduce the HDG formulation, we rewrite Equation (4) as a system of first-order equations by identifying $\mathbf{q} = -\mathbf{A}(p)\nabla p$ as the diffusive flux and $\mathbf{F}(p)$ as the convective flux

$$\begin{cases} \mathbf{q} + \mathbf{A}(p)\nabla p = 0 & \forall \mathbf{x} \in \Omega, \quad \forall t \in T, \\ s(p)\dfrac{\partial p}{\partial t} + \nabla \cdot (\mathbf{q} + \mathbf{F}(p)) = f(\mathbf{x},t) & \forall \mathbf{x} \in \Omega, \quad \forall t \in T, \\ p(\mathbf{x},t) = g_D(\mathbf{x},t) & \forall \mathbf{x} \in \Gamma_D, \quad \forall t \in T, \\ (\mathbf{q} + \mathbf{F}(p)) \cdot \mathbf{n} = g_N(\mathbf{x},t) & \forall \mathbf{x} \in \Gamma_N, \quad \forall t \in T, \\ p(\mathbf{x},0) = p^0(\mathbf{x}) & \forall \mathbf{x} \in \Omega. \end{cases} \tag{6}$$

3.2 Spatial discretization

We discretize the domain, Ω, with a tessellation, T_h, composed of a set of conformal elements of polynomial degree P. We define the following discontinuous finite element spaces associated to the tessellation T_h

$$\begin{aligned} \boldsymbol{\mathcal{W}}_h^P &= \left\{ \mathbf{w}_h \in L^2\left(\Omega^d\right) \mid \mathbf{w}_h|_e \in \left(\mathcal{S}^P\left(e\right)\right)^d \forall e \in \mathrm{T}_h \right\}, \\ \mathcal{V}_h^P &= \left\{ v_h \in L^2\left(\Omega^d\right) \mid v_h|_e \in \left(\mathcal{S}^P\left(e\right)\right) \forall e \in \mathrm{T}_h \right\}, \\ \mathcal{M}_h^P &= \left\{ \lambda_h \in L^2\left(\Sigma_h\right) \mid \lambda_h|_f \in \left(\mathcal{S}^P\left(f\right)\right) \forall f \in \Sigma_h \right\}, \end{aligned}$$

where e is an element of T_h, f is a face of the mesh skeleton, Σ_h, and $\mathcal{S}^P$ is the space of the polynomials of degree at most P for triangles and tetrahedra (usually denoted by $\mathcal{P}^P$), or the tensor products of polynomials of degree at most P in each coordinate direction for tensor product elements (usually denoted by $\mathcal{Q}^P$). In this work, we use a fixed polynomial degree for all the elements.

We define $\mathcal{M}_h^P\left(g_D\right) = \left\{ \lambda_h \in \mathcal{M}_h^P | \lambda_h = \Pi(g_D) \text{ on } \Gamma_D \right\}$, where $\Pi(\cdot)$ is a projection operator to the space $\left\{ \lambda_h|_{\Gamma_D} \, \forall \lambda_h \in \mathcal{M}_h^P \right\}$. We also consider the

scalar products

$$
\begin{aligned}
(\mathbf{a},\mathbf{b})_e &= \int_e \mathbf{a}\cdot\mathbf{b}\ \mathrm{d}\Omega \quad \forall \mathbf{a},\mathbf{b}\in \boldsymbol{\mathcal{W}}_h^P,\\
(a,b)_e &= \int_e a\,b\ \mathrm{d}\Omega \qquad \forall a,b\in \mathcal{V}_h^P,\\
\langle a,b\rangle_{\partial e} &= \int_{\partial e} a\,b\ \mathrm{d}\Gamma \qquad \forall a,b\in \mathcal{M}_h^P.
\end{aligned}
$$

From Equation (6), the HDG formulation ends up with finding an approximation $(\mathbf{q}_h, p_h, \hat{p}_h)\in \boldsymbol{\mathcal{W}}_h^P\times\mathcal{V}_h^P\times\mathcal{M}_h^P(g_D)$ such that

$$
\sum_e \left((\mathbf{A}^{-1}(p_h)\,\mathbf{q}_h,\mathbf{w}_h)_e - (p_h,\nabla\cdot\mathbf{w}_h)_e + \langle \hat{p}_h,\mathbf{w}_h\cdot\mathbf{n}\rangle_{\partial e}\right)=0, \tag{7a}
$$

$$
\begin{aligned}
\sum_e \left((s(p_h)\frac{\partial p_h}{\partial t},v_h)_e - (\mathbf{q}_h+\mathbf{F}(p_h),\nabla v_h)_e\right)+\\
\sum_e \left(\langle(\hat{\mathbf{q}}_h+\hat{\mathbf{F}}_h)\cdot\mathbf{n},\ v_h\rangle_{\partial e} - (f,v_h)_e\right)=0,
\end{aligned} \tag{7b}
$$

$$
\sum_e \left(\langle(\hat{\mathbf{q}}_h+\hat{\mathbf{F}}_h)\cdot\mathbf{n},\lambda_h\rangle_{\partial e}\right) - \langle g_N,\lambda_h\rangle_{\Gamma_N}=0, \tag{7c}
$$

for all $(\mathbf{w}_h, v_h, \lambda_h)\in\ \boldsymbol{\mathcal{W}}_h^P\times\mathcal{V}_h^P\times\mathcal{M}_h^P(0)$, where $\hat{p}_h$ is the trace of the pressure defined on the mesh skeleton, Σ_h, and $\hat{\mathbf{q}}_h+\hat{\mathbf{F}}_h$ is the total numerical flux. Equation (7c) is the transmissivity equation, in which we impose the continuity of the total numerical flux in the normal direction between adjacent elements. Therefore, this equation relates the unknowns between adjacent elements.

According to [35,36], we define the total numerical flux as

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

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

Following [35,36], we split the τ parameter into the diffusive and convective terms as

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

and we set the diffusive and convective numerical fluxes as:

$$
\hat{\mathbf{q}}_h = \mathbf{q}_h+\tau_{\mathrm{diff}}(p_h-\hat{p}_h)\mathbf{n}, \qquad \text{on } \Sigma_h, \tag{8a}
$$

$$
\hat{\mathbf{F}}_h = \mathbf{F}(\hat{p}_h)+\tau_{\mathrm{conv}}(p_h-\hat{p}_h)\mathbf{n}, \qquad \text{on } \Sigma_h, \tag{8b}
$$

respectively. We define the diffusive stabilization parameter, τ_{diff} as

$$
\tau_{\mathrm{diff}} = \frac{A}{l_c}, \tag{9}
$$

where l_c is a characteristic length of the problem, and

$$A = \frac{\rho(p_h)}{\mu}\gamma_{\mathbf{K}},$$

being $\gamma_{\mathbf{K}}$ the maximum eigenvalue of the permeability tensor, $\mathbf{K}$.

To select the τ_{conv} we use a monotone scheme flux, which ensures the stability of the numerical method [35,36]. Specifically, we define τ_{conv} as

$$\tau_{\text{conv}} = \frac{1}{(p_h - \hat{p}_h)^2}\int_{\hat{p}_h}^{p_h}\left(\frac{\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(s,\hat{p}_h) - \mathbf{F}(\hat{p}_h)\cdot\mathbf{n}}{p_h - \hat{p}_h}\right)\mathrm{d}s, \tag{10}$$

where $\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(\cdot,\cdot)$ is the Engquist-Osher monotone scheme flux

$$\hat{\mathbf{F}}\cdot\mathbf{n}^{EO}(a,b) = \frac{1}{2}\left(\mathbf{F}(a)+\mathbf{F}(b)\right)\cdot\mathbf{n} - \frac{1}{2}\int_a^b |\mathbf{F}'(s)\cdot\mathbf{n}|\mathrm{d}s, \tag{11}$$

see details in [35,36]. It is straightforward to prove that the derivative of the convective flux is

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

Therefore, the sign of the integral in Equation (11) only depends on the product $(\mathbf{Kg})\cdot\mathbf{n}$. Thus, inserting the third equality of Equation (5), and Equations (10) and (11) into Equation (8b), we obtain the normal component of the convective numerical flux, $\hat{\mathbf{F}}_h\cdot\mathbf{n}$, as

$$\hat{\mathbf{F}}_h\cdot\mathbf{n} = \begin{cases} \dfrac{(\mathbf{Kg})\cdot\mathbf{n}}{3c_f\rho_{ref}\mu}\left(\dfrac{\rho(p_h)^3 - \rho(\hat{p}_h)^3}{p_h - \hat{p}_h}\right) & \text{if } (\mathbf{Kg})\cdot\mathbf{n} \geq 0, \\ \dfrac{\rho(\hat{p}_h)^2\,(\mathbf{Kg})\cdot\mathbf{n}}{\mu} & \text{if } (\mathbf{Kg})\cdot\mathbf{n} < 0. \end{cases} \tag{12}$$

Note that for $p_h = \hat{p}_h$, and applying the Hôpital rule to the first equation of (12), we verify the required property of the monotone scheme flux: $\hat{\mathbf{F}}_h\cdot\mathbf{n}(p,p) = \mathbf{F}(p)\cdot\mathbf{n}$.

Afterwards, substituting the diffusive numerical flux, Equation (8a), into Equation (7) and using Equation (12) as the convective numerical flux, leads

to find $(\mathbf{q}_h, p_h, \hat{p}_h) \in \boldsymbol{\mathcal{W}}_h^P \times \mathcal{V}_h^P \times \mathcal{M}_h^P(g_D)$ such that

$$\sum_e \left((\mathbf{A}^{-1}\mathbf{q}_h, \mathbf{w}_h)_e - (p_h, \nabla \cdot \mathbf{w}_h)_e + \langle \hat{p}_h, \mathbf{w}_h \cdot \mathbf{n} \rangle_{\partial e} \right) = 0, \tag{13a}$$

$$\sum_e \left((s\frac{\partial p_h}{\partial t}, v_h)_e - (\mathbf{q}_h + \mathbf{F}_h, \nabla v_h)_e + \langle \hat{\mathbf{F}}_h \cdot \mathbf{n},\ v_h \rangle_{\partial e} \right) + \sum_e \left(\langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ v_h \rangle_{\partial e} - (f, v_h)_e \right) = 0, \tag{13b}$$

$$\sum_e \left(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n},\ \lambda_h \rangle_{\partial e} + \langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ \lambda_h \rangle_{\partial e} \right) - \langle g_N, \lambda_h \rangle_{\partial \Omega_N} = 0. \tag{13c}$$

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

Remark 1 The HDG method is conservative at the elemental level. This is deduced from the second equation in Equation (13), by setting the test function, $v_h = 1$, in a single element and 0 in the rest

$$\underbrace{(s\frac{\partial p_h}{\partial t}, 1)_e}_{\text{temporal variation}} + \underbrace{\langle (\hat{\mathbf{q}}_h + \hat{\mathbf{F}}_h) \cdot \mathbf{n},\ 1 \rangle_{\partial e}}_{\text{boundary inflow/outflow}} = \underbrace{(f, 1)_e}_{\text{source term}}, \tag{14}$$

that *represents* the mass conservation, Equation (1a) in integral form.

Let $\{N_i\}_{i=1,\ldots,N}$ be a Lagrangian basis of shape functions of $\mathcal{S}^P(e)$, where N is the total number of element nodes, and let $\{N_l^f\}_{l=1,\ldots,N_f}$ be a Lagrangian basis of shape functions of $\mathcal{S}^P(f)$, where N_f is the total number of nodes on the element faces. We define the approximations $\mathbf{q}_h$, p_h and $\hat{p}_h$ as

$$\mathbf{q}_h(\mathbf{x}, t) = \sum_{e \in \mathrm{T}_h} \sum_{i=1}^{N} \sum_{j=1}^{N_{sd}} q_{i,j}(t) N_i(\mathbf{x}) \mathbf{e}_j, \tag{15}$$

$$p_h(\mathbf{x}, t) = \sum_{e \in \mathrm{T}_h} \sum_{i=1}^{N} p_i(t) N_i(\mathbf{x}), \tag{16}$$

$$\hat{p}_h(\mathbf{x}, t) = \sum_{f \in \Sigma_h} \sum_{l=1}^{N_f} \hat{p}_l(t) N_l^f(\mathbf{x}), \tag{17}$$

where N_{sd} is the physical dimension of the problem. Similarly, the approximation of partial derivative respect time of the pressure, $\dot{p}_h$, is defined as

$$\dot{p}_h(\mathbf{x}, t) = \frac{\partial p_h(\mathbf{x}, t)}{\partial t} = \sum_{e \in \mathrm{T}_h} \sum_{i=1}^{N} \dot{p}_i(t) N_i(\mathbf{x}). \tag{18}$$

Inserting Equations (15), (16), (17) and (18) into Equation (13), we obtain a non-linear coupled system of first order DAE. Specifically, the problem consists of finding the coefficients $q_{i,j}(t)$, $p_i(t)$, $\dot{p}_i(t)$ for $i = 1 \dots N_i$, $j = 1 \dots N_{sd}$ and $\hat{p}_l(t)$ for $l = 1 \dots N_f$

$$\sum_e \left((\mathbf{A}^{-1}\mathbf{q}_h, N_i\mathbf{e}_j)_e - (p_h, \nabla \cdot (N_i\mathbf{e}_j))_e + \langle \hat{p}_h, N_i\mathbf{e}_j \cdot \mathbf{n} \rangle_{\partial e} \right) = 0 \tag{19a}$$

$$\begin{aligned} \sum_e \left((s\dot{p}_h, N_i)_e - (\mathbf{q}_h + \mathbf{F}_h, \nabla N_i)_e + \langle \hat{\mathbf{F}}_h \cdot \mathbf{n},\ N_i \rangle_{\partial e} \right) \\ + \sum_e \left(\langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ N_i \rangle_{\partial e} - (f, N_i)_e \right) = 0 \end{aligned} \tag{19b}$$

$$\sum_e \left(\langle \hat{\mathbf{F}}_h \cdot \mathbf{n},\ N_l^f \rangle_{\partial e} + \langle \mathbf{q}_h \cdot \mathbf{n} + \tau_{\text{diff}}(p_h - \hat{p}_h),\ N_l^f \rangle_{\partial e} \right) - \langle g_N, N_l^f \rangle_{\partial \Omega_N} = 0 \tag{19c}$$

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

3.3 Temporal discretization

Equation (19) is a DAE and we rewrite it as

$$\mathbf{R}\left(t, \mathbf{q}, \mathbf{p}, \dot{\mathbf{p}}, \hat{\mathbf{p}}\right) = \begin{bmatrix} \mathbf{R}_q\left(t, \mathbf{q}, \mathbf{p}, \dot{\mathbf{p}}, \hat{\mathbf{p}}\right) \\ \mathbf{R}_p\left(t, \mathbf{q}, \mathbf{p}, \dot{\mathbf{p}}, \hat{\mathbf{p}}\right) \\ \mathbf{R}_{\hat{p}}\left(t, \mathbf{q}, \mathbf{p}, \dot{\mathbf{p}}, \hat{\mathbf{p}}\right) \end{bmatrix} = \mathbf{0}, \tag{20}$$

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

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

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

To solve the DAE in Equation (20), we use a diagonally implicit Runge-Kutta method (DIRK). From now on, we denote by $(\cdot)^n$ the value of any variable at time t^n, and by $(\cdot)^{n,i}$ the value of any variable at time $t^{n,i} = t^n + c_i \Delta t$, being n the time step and i the DIRK stage. According t theses schemes, we compute the pressure at time $t^{n+1} = t^n + \Delta t$ as

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \Delta t \sum_{i=1}^{s} b_i \dot{\mathbf{p}}^{n,i}, \tag{21}$$

Table 1: Butcher's table for a diagonal implicit Runge-Kutta scheme.

c_1	a_{11}			
c_2	a_{21}	a_{22}		
$\vdots$	$\vdots$		$\ddots$	
c_s	a_{s1}		$\dots$	a_{ss}
	b_1	b_2	$\dots$	b_s

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

$$\mathbf{p}^{n,i} = \mathbf{p}^n + \Delta t \sum_{j=1}^{i} a_{ij} \dot{\mathbf{p}}^{n,j}. \tag{22}$$

Therefore, the pressure at t^{n+1} and the pressure at all the stages, $\dot{\mathbf{p}}^{n,i}$ for $i = 1, \dots, s$, can be computed once the approximation to the derivatives of the pressures, $\dot{\mathbf{p}}^{n,i}$, is known at all the stages $i = 1, \dots, s$.

To compute these approximations, we insert Equation (22) into the DAE (20) and obtain

$$\mathbf{R}\left(t^{n,i}, \mathbf{q}^{n,i}, \mathbf{p}^n + \Delta t \sum_{j=1}^{i} a_{ij} \dot{\mathbf{p}}^{n,j}, \dot{\mathbf{p}}^{n,i}, \hat{\mathbf{p}}^{n,i}\right) = \mathbf{0} \qquad \text{for } i = 1, \dots, s. \tag{23}$$

Equation (23) can be rewritten as

$$\mathbf{R}\left(t^{n,i}, \mathbf{q}^{n,i}, \mathbf{p}^n + \Delta t \sum_{j=1}^{i-1} a_{ij} \dot{\mathbf{p}}^{n,j} + \Delta t a_{ii} \dot{\mathbf{p}}^{n,i}, \dot{\mathbf{p}}^{n,i}, \hat{\mathbf{p}}^{n,i}\right) = \mathbf{0} \qquad \text{for } i = 1, \dots, s \tag{24}$$

to highlight that the unknowns at the i-th stage are the approximation to the derivative, $\dot{\mathbf{p}}^{n,i}$, the fluxes, $\mathbf{q}^{n,i}$, and the traces $\hat{\mathbf{p}}^{n,i}$. Note that the pressure at the stages, $\mathbf{p}^{n,i}$ for $i = 1, \dots, s$, is not an unknown because we have used Equation (22) to write the pressure in terms of its temporal derivative. In addition, the time derivative of the pressure at the previous stages $\mathbf{p}^{n,j}$ for $j = 1, \dots, i-1$, is not an unknown since they have been already computed. Finally, once the approximations to the derivatives $\dot{\mathbf{p}}^{n,i}$, for $i = 1, \dots, s$, are computed, the pressure at next time step is computed according to Equation (21).

The parameters b_i, c_i, a_{ij}, with $i = 1 \dots s$ and $j = 1 \dots i$, define the DIRK method, and are given by Butcher's tables, see Table 1, [7,33,23].

3.4 Non-linear solver

Equation (24) defines a non-linear system of equations that we solve using the Newton-Raphson method. From now on and without loss of generality, we

reorder the unknowns of the non-linear system of Equation (24) such as

$$\mathbf{u} = \begin{bmatrix} \mathbf{q}^i \\ \dot{\mathbf{p}}^i \\ \hat{\mathbf{p}}^i \end{bmatrix}.$$

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

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

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

$$\mathbf{J}\left(\mathbf{u}^{i,k}\right)\delta\mathbf{u}^{i,k} = -\mathbf{R}\left(\mathbf{u}^{i,k}\right), \tag{25}$$

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

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

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

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

3.5 Hybridization procedure

To reduce the computational cost, we hybridize the linear system in Equation (25) using the static condensation procedure that allows solving only for the unknowns, $\delta\hat{\mathbf{p}}^{i,k}$. Then, $\delta\mathbf{q}^{i,k}$ and $\delta\dot{\mathbf{p}}^{i,k}$ are obtained using an element-by-element post-process. To this end, we rewrite the linear system in Equation (25) as

$$\begin{bmatrix} \mathbf{J}_{qq}^{i,k} & \mathbf{J}_{q\dot{p}}^{i,k} & \mathbf{J}_{q\hat{p}}^{i,k} \\ \mathbf{J}_{\dot{p}q}^{i,k} & \mathbf{J}_{\dot{p}\dot{p}}^{i,k} & \mathbf{J}_{\dot{p}\hat{p}}^{i,k} \\ \mathbf{J}_{\hat{p}q}^{i,k} & \mathbf{J}_{\hat{p}\dot{p}}^{i,k} & \mathbf{J}_{\hat{p}\hat{p}}^{i,k} \end{bmatrix} \begin{bmatrix} \delta\mathbf{q}^{i,k} \\ \delta\dot{\mathbf{p}}^{i,k} \\ \delta\hat{\mathbf{p}}^{i,k} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_q^{i,k} \\ \mathbf{R}_{\dot{p}}^{i,k} \\ \mathbf{R}_{\hat{p}}^{i,k} \end{bmatrix}. \tag{26}$$

Afterwards, we split Equation (26) as

$$\begin{bmatrix} \mathbf{J}_{qq}^{i,k} & \mathbf{J}_{q\dot{p}}^{i,k} \\ \mathbf{J}_{\dot{p}q}^{i,k} & \mathbf{J}_{\dot{p}\dot{p}}^{i,k} \end{bmatrix} \begin{bmatrix} \delta\mathbf{q}^{i,k} \\ \delta\dot{\mathbf{p}}^{i,k} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_{q\hat{p}}^{i,k} \\ \mathbf{J}_{\dot{p}\hat{p}}^{i,k} \end{bmatrix} \delta\hat{\mathbf{p}}^{i,k} = - \begin{bmatrix} \mathbf{R}_q^{i,k} \\ \mathbf{R}_{\dot{p}}^{i,k} \end{bmatrix} \tag{27a}$$

$$\begin{bmatrix} \mathbf{J}_{\hat{p}q}^{i,k} & \mathbf{J}_{\hat{p}\dot{p}}^{i,k} \end{bmatrix} \begin{bmatrix} \delta\mathbf{q}^{i,k} \\ \delta\dot{\mathbf{p}}^{i,k} \end{bmatrix} + \mathbf{J}_{\hat{p}\hat{p}}^{i,k}\delta\hat{\mathbf{p}}^{i,k} = -\mathbf{R}_{\hat{p}}^{i,k}, \tag{27b}$$

From Equation (27a), we obtain

$$\begin{bmatrix}\delta\mathbf{q}^{i,k}\\ \delta\dot{\mathbf{p}}^{i,k}\end{bmatrix} = \begin{bmatrix}\mathbf{J}^{i,k}_{qq} & \mathbf{J}^{i,k}_{q\dot{p}}\\ \mathbf{J}^{i,k}_{\dot{p}q} & \mathbf{J}^{i,k}_{\dot{p}\dot{p}}\end{bmatrix}^{-1}\left(-\begin{bmatrix}\mathbf{R}^{i,k}_{q}\\ \mathbf{R}^{i,k}_{\dot{p}}\end{bmatrix} - \begin{bmatrix}\mathbf{J}^{i,k}_{q\hat{p}}\\ \mathbf{J}^{i,k}_{\dot{p}\hat{p}}\end{bmatrix}\delta\hat{\mathbf{p}}^{i,k}\right). \tag{28}$$

Then, we substitute Equation (28) into Equation (27b), and we obtain $\delta\hat{\mathbf{p}}^{i,k}$ as the solution of the hybridized linear system

$$\begin{aligned}\left(-\begin{bmatrix}\mathbf{J}^{i,k}_{\hat{p}q} & \mathbf{J}^{i,k}_{\hat{p}\dot{p}}\end{bmatrix}\begin{bmatrix}\mathbf{J}^{i,k}_{qq} & \mathbf{J}^{i,k}_{q\dot{p}}\\ \mathbf{J}^{i,k}_{\dot{p}q} & \mathbf{J}^{i,k}_{\dot{p}\dot{p}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{J}^{i,k}_{q\hat{p}}\\ \mathbf{J}^{i,k}_{\dot{p}\hat{p}}\end{bmatrix} + \mathbf{J}^{i,k}_{\hat{p}\hat{p}}\right)\delta\hat{\mathbf{p}}^{i,k} =\\ -\mathbf{R}^{i,k}_{\hat{p}} + \begin{bmatrix}\mathbf{J}^{i,k}_{\hat{p}q} & \mathbf{J}^{i,k}_{\hat{p}\dot{p}}\end{bmatrix}\begin{bmatrix}\mathbf{J}^{i,k}_{qq} & \mathbf{J}^{i,k}_{q\dot{p}}\\ \mathbf{J}^{i,k}_{\dot{p}q} & \mathbf{J}^{i,k}_{\dot{p}\dot{p}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{R}^{i,k}_{q}\\ \mathbf{R}^{i,k}_{p}\end{bmatrix},\end{aligned} \tag{29}$$

Finally, using $\delta\hat{\mathbf{p}}^{i,k}$, we compute $\delta\mathbf{q}^{i,k}$ and $\delta\dot{\mathbf{p}}^{i,k}$ by solving the linear system in Equation (28). It is important to highlight that Equation (28) is solved element by element, since we can reorder the terms of the matrix

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

to convert it into a block diagonal matrix, where each block involves only unknowns of a single element. Thus, the computational cost of solving Equation (28) is low and this process can be easily parallelized.

To solve the global system, Equation (29) at each stage of the Runge-Kutta method, $i = 1, \dots, s$, and at each Newton-Raphson iteration, k, we use LU factorization.

3.6 Local post-processing

One of the main advantages of using the HDG formulation is that the pressure, p_h, and its flux, $\mathbf{q}_h$, in $\mathcal{V}_h^P$ and $\boldsymbol{\mathcal{W}}_h^P$ spaces, respectively, have a rate of convergence of $P+1$ in the L^2-norm, when the temporal error is low enough. Moreover, a local post-processing can be applied to obtain a new approximation for the pressure, p_h^*, in $\mathcal{V}_h^{P+1}$ with a rate of convergence of $P+2$ in the L^2-norm for diffusion dominated problems, see [25,36].

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

$$(\mathbf{A}(p_h)\nabla p_h^*, \nabla v_h)_e = -(\mathbf{q}_h, \nabla v_h)_e, \tag{30a}$$

$$(p_h^*, 1)_e = (p_h, 1)_e, \tag{30b}$$

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

In order to obtain a well-posed and invertible system, Equation (30b) is added, which imposes that the averages of the post-processed pressure, p_h^*, and the approximated pressure, p_h, are equal element by element. According to [25,36] it is important to highlight that this procedure can be applied at selected time steps, and it is not necessary to apply it to all the time steps.

4 Examples

This section presents several examples that illustrate the capabilities of the proposed high-order HDG formulation coupled with DIRK schemes. The first example shows numerical evidence of the optimal convergence rates of $P+1$ in L^2-norm for the pressure, the flux and the Darcy velocity, and also the convergence rate of $P+2$ in L^2-norm for the post-processed pressure. In addition, we check the elemental mass balance. The second example is devoted to validate our formulation and its implementation by comparing the obtained solution against an analytical approximation. Moreover, we compare the accuracy of the obtained approximations using different temporal integration schemes and different time steps for each scheme. The third example shows the flow through a highly heterogeneous porous medium. We also compare the accuracy of the pressure, flux and velocity using different polynomial degrees. The last example is a three dimensional case with three permeability regions, in which we consider isotropic and anisotropic diagonal permeability tensors.

For all the examples, we set $p_{ref} = p(\mathbf{x}, 0) := p^0$. All the high-order meshes are generated using the algorithms presented in [17,42,16] that are implemented in the EZ4U environment [41]. We set the number of integration points as $2P + 1$ per direction. To perform the temporal integration, we use the DIRK schemes defined in Appendix B. In addition, the stopping tolerances for the non-linear system are

$$\varepsilon_{\mathbf{q}} = \varepsilon_{\dot{p}} = \varepsilon_{\hat{p}} = 10^{-7},$$
$$\varepsilon_{\mathbf{F_q}} = \varepsilon_{\mathbf{F}_{\dot{p}}} = \varepsilon_{\mathbf{F}_{\hat{p}}} = 10^{-5}.$$

4.1 Convergence rate analysis

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

$$p = (1 + \sin(2\pi x)\sin(2\pi y)\sin(0.25\pi t)), \tag{31}$$

where $\mathbf{x} = (x, y) \in (0, 1) \times (0, 1)$m and $t \in [0, 1]$s. We prescribe a Dirichlet boundary condition on the whole boundary and a source term in order to obtain the analytical pressure defined in Equation (31). The material and fluid parameters used in this example are detailed in Table 2.

We generate a series of meshes of quadrilateral elements of polynomial degree between two to five. We perform the temporal integration using a DIRK3-s3 scheme with an appropriated Δt to balance the temporal and spatial errors. Then, we measure the error in L^2-norm of the obtained approximations against the analytical solution at time 1s.

Figures 1(a) and 1(b) show the convergence rate of the pressure and the flux in L^2-norm, respectively. We obtain the theoretically expected convergence rate of $P+1$ in L^2-norm for all the cases. Figure 1(c) shows the obtained

Table 2: Material and fluid parameters for Example 4.1.

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

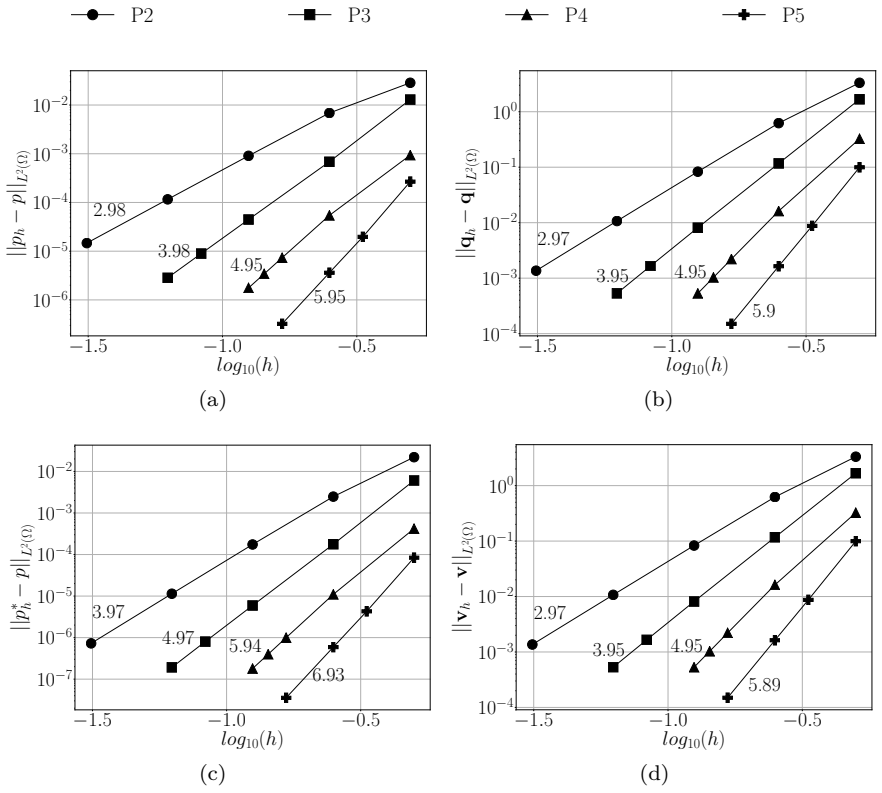


Fig. 1: Convergence rate for: a) the pressure; b) the flux, c) the post-processed pressure and d) the Darcy velocity.

L^2-error for the post-processed pressure, p_h^*, and also shows the expected convergence rates of $P+2$ for all the cases. Moreover, we also obtain a convergence rate of $P+1$ in L^2-norm for the velocity, because it is defined in terms of the pressure and its flux, see Figure 1(d).

For all the meshes, we have checked the elemental mass balance, see Equation (14). We obtain values in the range $6 \cdot 10^{-10}$ to $6 \cdot 10^{-14}$. For instance, Figure 2 shows the elemental mass balance for the mesh with 16 elements and polynomial degree five.

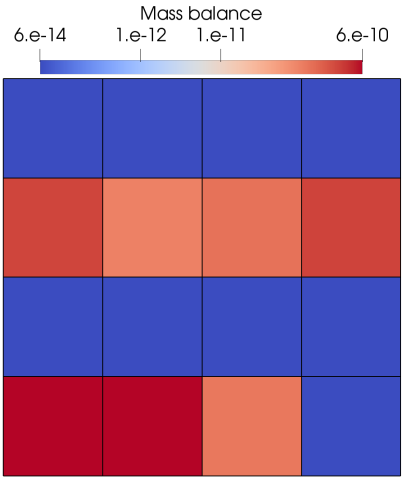


Fig. 2: Elemental mass balance for the mesh with 16 elements and polynomial degree five.

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

4.2 Comparison with an analytical approximation

In this example, we compare the solution provided by our formulation with an analytical approximation for a one dimensional radial flow, see [8]. To this end, and according to [8], we assume an infinite horizontal and homogeneous reservoir with constant material properties and a punctual and isolated well. Neglecting the gravity effects and using cylindrical coordinates, Equation (3) is expressed as

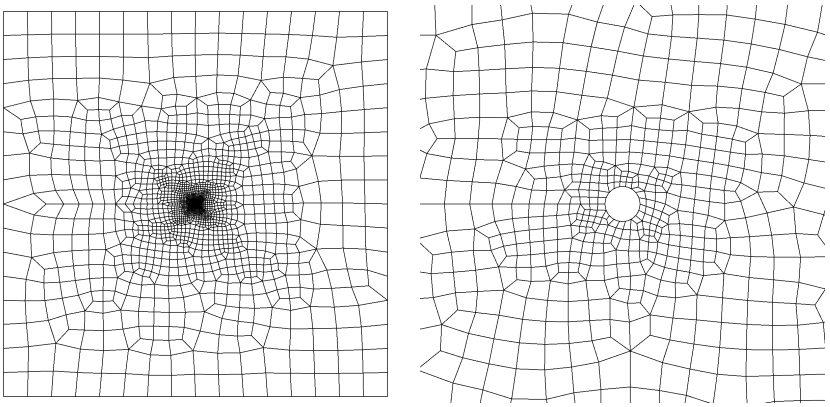
$$\frac{1}{\chi}\frac{\partial p}{\partial t} = \frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r}, \tag{32}$$

where r is the distance to the punctual well and $\chi = k/\phi\mu c_t$. The PDE in Equation (32) is completed with the following boundary and initial conditions

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

Table 3: Material and fluid parameters for Example 4.2.

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



(a) (b)

Fig. 3: Unstructured mesh for the considered reservoir; a) global view and b) detailed view near the well.

where Q is the oil production well rate and H is the reservoir thickness. Under these conditions, the analytical approximation of Equation (3) is

$$p\left(r,t\right) = p^0 - \frac{Q\mu}{2\pi kH} ln\left(\frac{2.25t\chi}{r^2}\right), \qquad t > 0, \tag{34}$$

see [8] for more details. This analytical approximation is valid when $r^2/4t\chi < 0.001$.

We consider a reservoir thickness of $H = 30.48$ m and an isolated well with radius $r_w = 5.715$ cm. We use the material and fluid parameters detailed in Table 3, see [8] for more details.

For the numerical simulation, we define a square domain $\Omega = (0, 8000) \times (0, 8000)$ m, with a circular hole at the center with radius $r_w = 5.715$ cm. We prescribe the pressure on the square boundary, Γ_D, and a fixed oil rate at the circular boundary, Γ_W.

The problem to be solved numerically is modeled by Equation (3) with a null source term, neglecting the gravitational effects, and the following bound-

ary and initial conditions

$$\begin{cases} \phi c_t \dfrac{\partial p}{\partial t} = \nabla \cdot \left(\dfrac{1}{\mu} \mathbf{K} \nabla p \right) & \forall \mathbf{x} \in \Omega, \forall t \in T \\ p\left(\mathbf{x}, t\right) = 244.966 \text{ atm} & \forall \mathbf{x} \in \partial\Gamma_D, \forall t \in \ T, \\ \left(\dfrac{1}{\mu} \mathbf{K} \nabla p \right) \cdot \mathbf{n} = \dfrac{Q\mu}{2\pi r k H} = 0.0527 \dfrac{\text{Kg}}{\text{m}^3 s} & \forall \mathbf{x} \in \partial\Gamma_W, \forall t \in \ T, \\ p\left(\mathbf{x}, 0\right) = 244.966 \text{ atm} & \forall \mathbf{x} \in \Omega, \end{cases}$$

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

We discretize the domain using an unstructured mesh of 4652 unstructured quadrilateral elements of polynomial degree three, see Figures 3(a) and 3(b). The total number of nodes is 41991 and the total unknowns of the linear system is 260676. However, after applying the hybridization procedure described in Section 3.5, the size of the linear system to be solved is reduced to 37380.

We simulate a total of four days of oil extraction using the backward Euler, the DIRK2-s2 and the DIRK4-s6 schemes with time steps $\Delta t = 0.8, 0.4, 0.2$ and 0.1 days. Specifically, we compare the relative error of the obtained approximations respect to the analytical solution at a point located on the boundary of the well, $\mathbf{x}_p = (4000.0715, 4000)$m, as:

$$E_p(t) = \frac{|p_h(\mathbf{x}_p, t) - p(\mathbf{x}_p, t)|}{|p(\mathbf{x}_p, t)|}$$

Figure 4 shows the obtained relative errors using the different temporal schemes with the selected time steps for all the temporal discretizations schemes. In all cases, the relative error decreases when using smaller time steps. Furthermore, using higher-order temporal schemes with the same time step, the relative error is smaller and decreases faster as time advances than using lower-order schemes. Note that for all the temporal integration schemes and time steps, the major difference between the analytical approximation and the numerical approximation appears at the first time step. This happens because the initial condition is not compatible with the boundary condition, in which the hydrocarbon is totally still and the well starts to pump at the prescribed flux rate. Afterwards, the numerical approximation tends to the analytical approximation as time advances.

The minimum relative error is $4.16 \cdot 10^{-7}$ that corresponds to the error introduced by the spatial discretization, see Figure 4. This example shows that to balance the temporal and spatial errors using low-order temporal schemes, small time steps are necessary. Even in the case of the backward Euler with the smallest time step, the obtained error at the last time step is around one order of magnitude larger than the spatial error. On the contrary, when using the DIRK4s6 scheme we can use a large time step to obtain similar spatial and temporal errors.

We obtain a similar relative error using the DIRK2s2 scheme with $\Delta t = 0.1$ days and the DIRK4s6 scheme with $\Delta t = 0.8$ days at time four days, see Figure

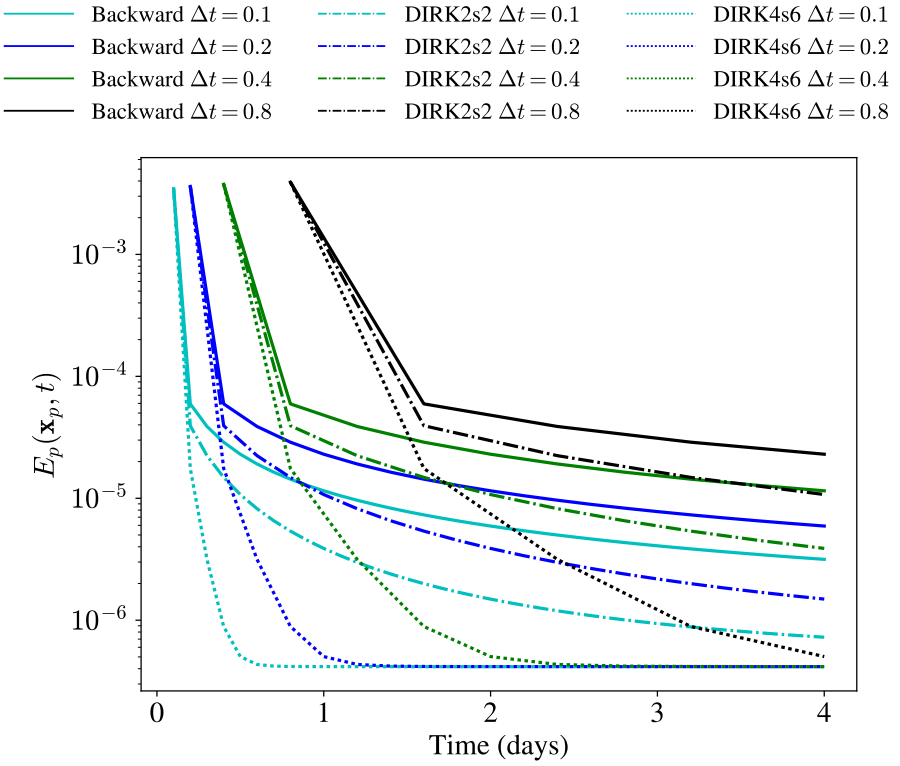


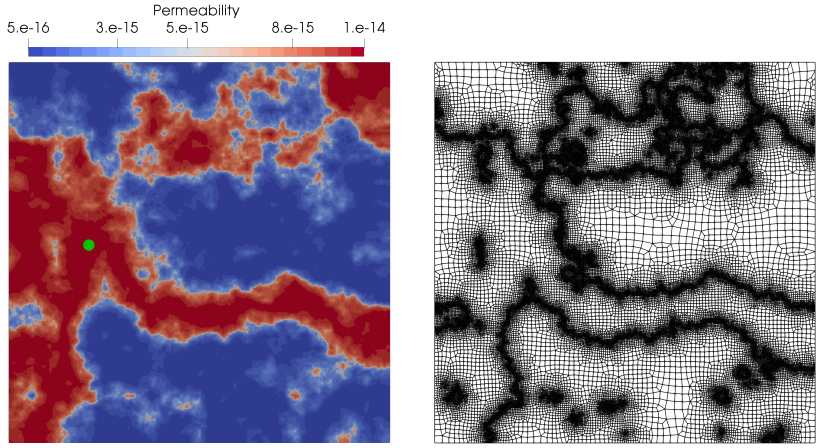
Fig. 4: Relatives errors against an analytical solution.

4. Nevertheless, the DIRK2s2 scheme with $\Delta t = 0.1$ days requires 40 steps, which leads to a total of 80 RK stages at four days. Whereas, the DIRK4s6 with $\Delta t = 0.8$ days only requires 5 steps, which leads to a total of 30 RK stages at the same four days. Thus, since the number of iterations of the Newton-Raphson method using both schemes is similar, the DIRK4s6 scheme is $80/30 = 2.6$ times faster than the DIRK2s2 even though it has four stages more per time step.

In this example, we have shown that we obtain similar accuracy with less computational cost using higher-order temporal schemes with larger time steps than using low-order temporal schemes with smaller time steps.

4.3 Flow through a highly heterogeneous medium

In this example, we perfom a simulation through a highly heterogeneous porous medium, see Figure 5(a). The permeability ranges between $5 \cdot 10^{-16}$ and 10^{-14}, and the other material parameters are detailed in Table 4.



(a) (b)

Fig. 5: a) Highly heterogeneous domain and permeability field, well location (green circle) and b) high-order unstructured mesh.

Table 4: Material and fluid parameters for the highly heterogeneous porous medium

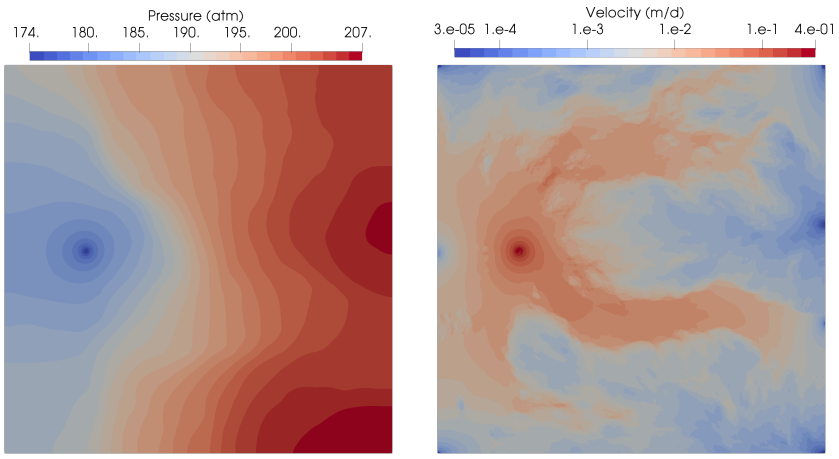
Parameter	Value	Parameter	Value
$\mathbf{K}_{min}$	$5 \cdot 10^{-16}$ m^2	$\mathbf{K}_{max}$	10^{-14} m^2
p^0	244.966 atm	ϕ_{ref}	0.2
c_r	$5.8 \cdot 10^{-10}$ Pa^{-1}	μ	0.00106 Pa $\cdot$ s
c_f	$1.45 \cdot 10^{-9}$ Pa^{-1}	ρ_{ref}	897.5 kg/m^3

The physical domain is $\Omega = (0, 100) \times (0, 100)$m, in which we impose no-flux condition on all the boundaries. In addition, we consider a source term defined as

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

being $f = -0.015$ kg/s, located at $\mathbf{x}_w = (21.0, 52.0)$ m and $r_w = 0.45$m, see Figure 5(a).

We discretize the domain with 59111 unstructured quadrilateral elements with polynomial degree from two to five. We adapt the element size to the Hessian of the permeability. That is, we have used smaller elements in the regions with higher curvature of the permeability field and larger elements in the regions with lower curvature, see Figure 5(b). To perform the temporal integration, we use DIRK3-s3 scheme with $\Delta t = 0.2$ days and the total simulation time is 4 days.



(a) (b)

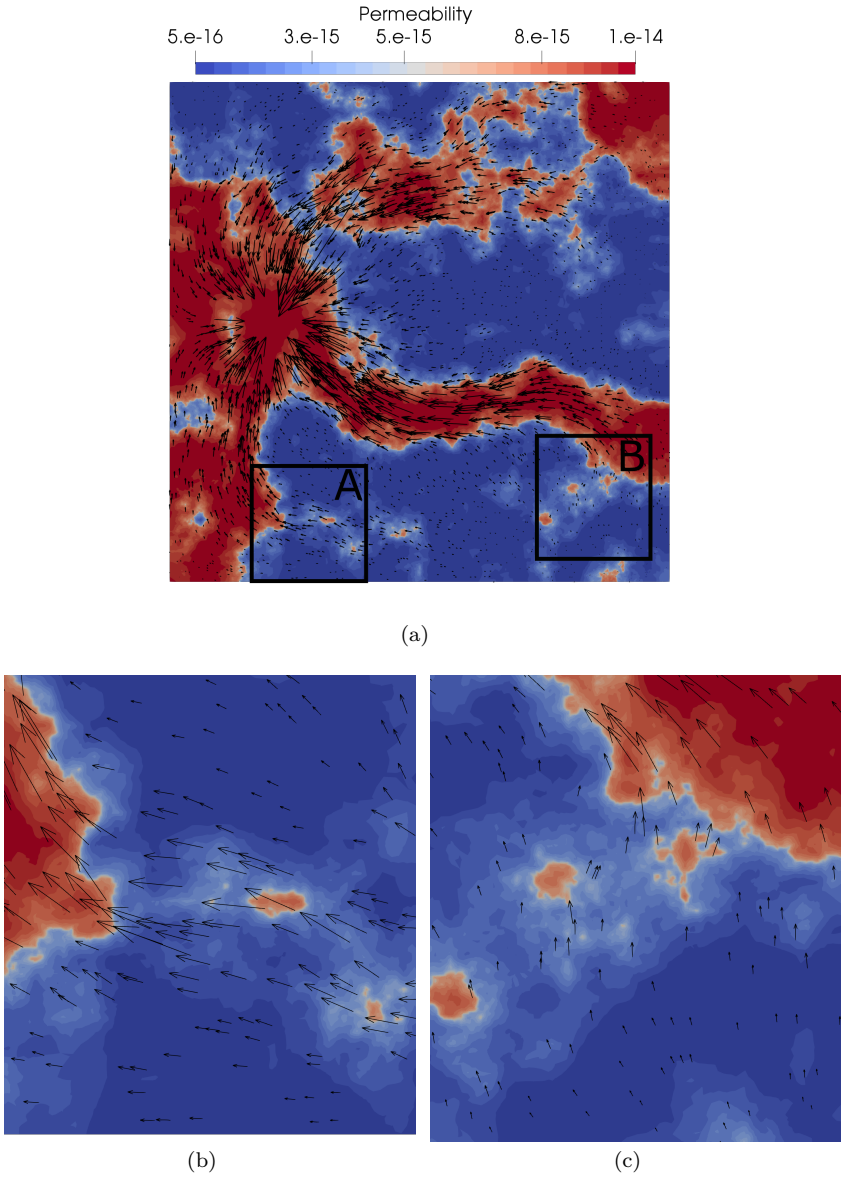
Fig. 6: Numerical solution at t=4 days. a) pressure and b) modulus of the velocity.

Figures 6(a) and 6(b) show the pressure and the modulus of the velocity at $t = 4$ days approximated with polynomials of degree five, respectively. The lowest pressure and the highest velocity values are around the well. Furthermore, the fluid is moving faster through the most permeable regions, see Figure 6(b). This effect can be also observed in Figure 7(a), which plots the velocity vectors and the permeability field. The velocity vectors are larger in the most permeable regions (red regions) than in the lowest ones (blue regions). Moreover, the fluid tends to move towards the most permeable regions, see Figures 7(b) and 7(c). Afterwards, the fluid moves through the most permeable regions.

We compute a reference solution for the pressure, the flux and the velocity using polynomials of degree five. Next, we compute the relative errors of the solutions obtained with polynomials of degree two to four respect to the reference solutions as:

$$E_p(t) = \frac{||p_h(\mathbf{x},t) - p_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}{||p_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}$$
$$E_{\mathbf{q}}(t) = \frac{||\mathbf{q}_h(\mathbf{x},t) - \mathbf{q}_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}{||\mathbf{q}_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}$$
$$E_{\mathbf{v}}(t) = \frac{||\mathbf{v}_h(\mathbf{x},t) - \mathbf{v}_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}{||\mathbf{v}_h^{P5}(\mathbf{x},t)||_{L^2(\Omega)}}$$

Figure 8 plots these relative errors for the three meshes. Note that as we increase the interpolation degree not only the scalar variable, but also the fluxes and velocities monotonically converge to the reference solution.



(a)

(b)

(c)

Fig. 7: Permeability field and velocity vectors at t=4 days. a) General view, b) detail of the lower-left region A, c) detail of the lower-right region B.

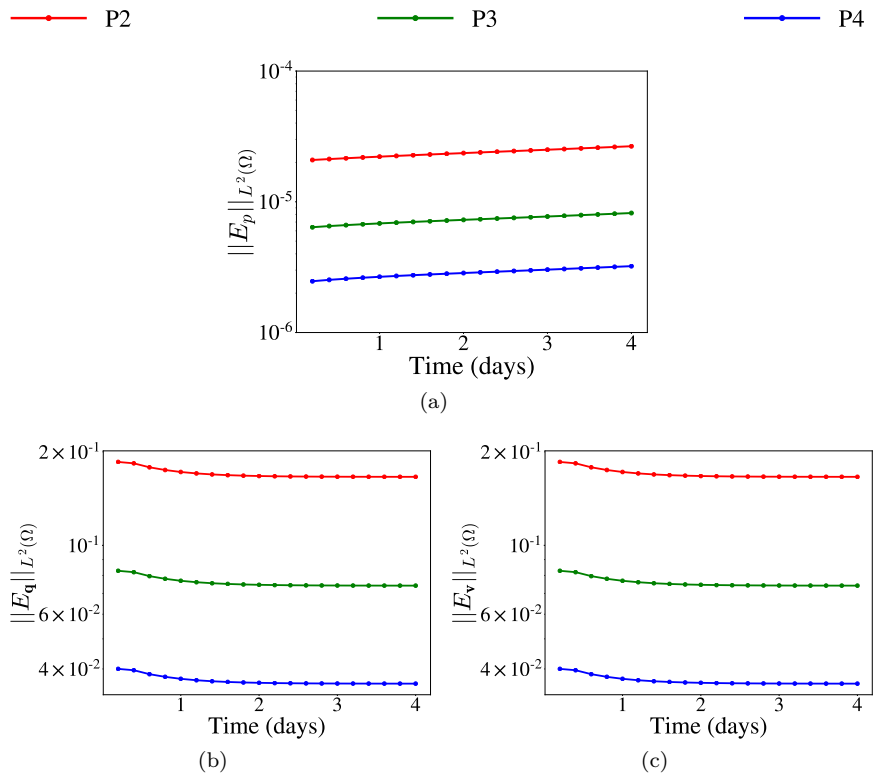


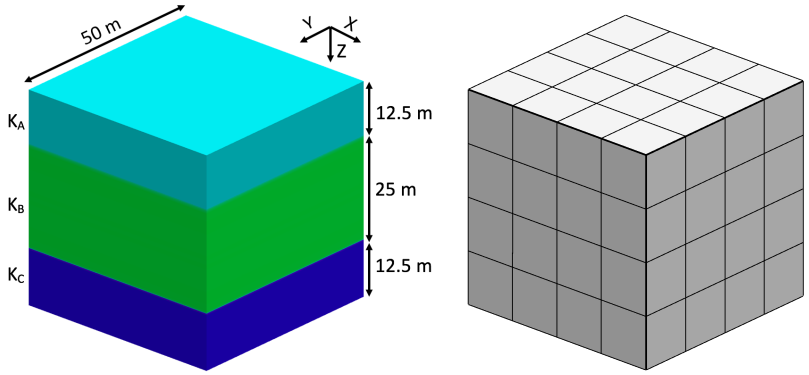
Fig. 8: Relative errors for: a) the pressure; b) the flux, and c) the velocity.

4.4 Three-dimensional flow through a heterogeneous anisotropic material

The last example corresponds to a fully three dimensional case with three different permeability regions, and for one of them we consider isotropic and anisotropic diagonal permeability tensors, see Figure 9(a). The most permeable region is located at the middle, $\mathbf{K}_B$. At the bottom is the region with the lowest permeability, $\mathbf{K}_C$. The upper region has an intermediate permeability value, $\mathbf{K}_A$. Specifically, we compare the pressure drop because of the hydrocarbon recovery considering isotropic and anisotropic diagonally permeability tensor in the middle region, $\mathbf{K}_B$. The permeability values for each region are:

$$\begin{aligned}
&\mathbf{K}_A: \quad k_{xx} = k_{yy} = k_{zz} = 10^{-14}\mathrm{m}^2,\\
&\mathbf{K}_B: k_{xx} = k_{zz} = 10^{-13}\mathrm{m}^2, k_{yy} = \alpha k_{zz},\\
&\mathbf{K}_C: \quad k_{xx} = k_{yy} = k_{zz} = 10^{-17}\mathrm{m}^2,
\end{aligned}$$

where $\alpha = 1$ for the isotropic case and $\alpha = 5$ for the anisotropic case. The other material parameters are detailed in Table 5



(a) (b)

Fig. 9: a) Permeability distribution. b) Hexahedral elements of polynomial degree four and with an element size of 12.5 m.

Table 5: Material and fluid parameters for Example 4.4.

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

The physical domain is $\Omega = (0, 50) \times (0, 50) \times (0, 50)$m, and we impose no-flow condition on all the boundaries. The extraction well is located in the middle region and is modeled using a source term as:

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

where $\mathbf{x}_w = (25, 25, 25)$m, $r_w = 4.0$m and $f = -2$kg/s.

We discretize the domain with a structured hexahedral mesh of 64 elements of polynomial degree three (2197 nodes), see Figure 9(b). This is a coarse mesh with four high-order elements at each edge of the cube. The total number of unknowns of the linear system is 20224. However, after applying the hybridization procedure, the size of the linear system to be solved is reduced to 3840. The time step for this simulation is $\Delta t = 0.1$ days.

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

$$\frac{\int_\Omega \|p^{n+1} - p^n\|^2 \mathrm{d}\Omega}{\int_\Omega 1 \mathrm{d}\Omega} < \varepsilon_{abs}, \tag{35}$$

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

Figure 10 shows the pressure field at time $t = 1$ day in two sections of the domain for the isotropic and the anisotropic permeability tensors. For both cases, the pressure increases with the depth because of the gravity effects, and it is lower near the source term, because of the hydrocarbon recovery. In addition, the pressure field in the impermeable zone remains higher than in the other two regions because of the low permeability value. In the other two regions the effect of the pumping well is negligible. Moreover, we plot the velocity vectors for both cases. Again, the different permeability of the regions affect the fluid velocity. In the middle region, which has the highest permeability, $\mathbf{K}_B$, the fluid is moving faster than in the upper region, where the permeability is lower, $\mathbf{K}_A$, while the velocity is almost zero in the impermeable region, $\mathbf{K}_C$.

In Figure 10, we also observe the differences between the isotropic and anisotropic permeability of the middle region, $\mathbf{K}_B$. For the anisotropic case, the fluid tends to move faster in the y-direction, which is five times more permeable than the the other directions. For that reason, the pressure drop in z-direction is higher if the permeability is isotropic, Figure 10(a), than the anisotropic one, Figure 10(c). Moreover, in the plane XY (perpendicular to the depth) the isobars have circular symmetry centered at the well for the isotropic permeability case, Figure 10(b), whereas the isobars for the anisotropic permeability has an ellipsoid shape, Figure 10(d). Finally, the velocity vectors are larger in the y-direction for the anisotropic permeability case since the fluid is moving faster in this direction, Figure 10(d).

5 Conclusions

In this work, we have presented two contributions. In the first contribution, we have developed a high-order HDG formulation combined with high-order DIRK schemes for the one-phase flow problem through porous media. To this end, we have rewritten the initial second-order PDE as a set of first-order PDE's, and the weak form of the problem has been deduced. In the second contribution, we have extended the work in [36] to deduce an analytical expression for the stabilization parameter of the proposed HDG formulation. We have split the stabilization parameter into diffusive and convective parts. The diffusive part is selected according to the physical values of the problem, and we have used the Engquist-Osher monotone scheme flux for the convective part. These choices ensure the existence and uniqueness of the obtained approximation while providing a stable and convergent method. We have shown

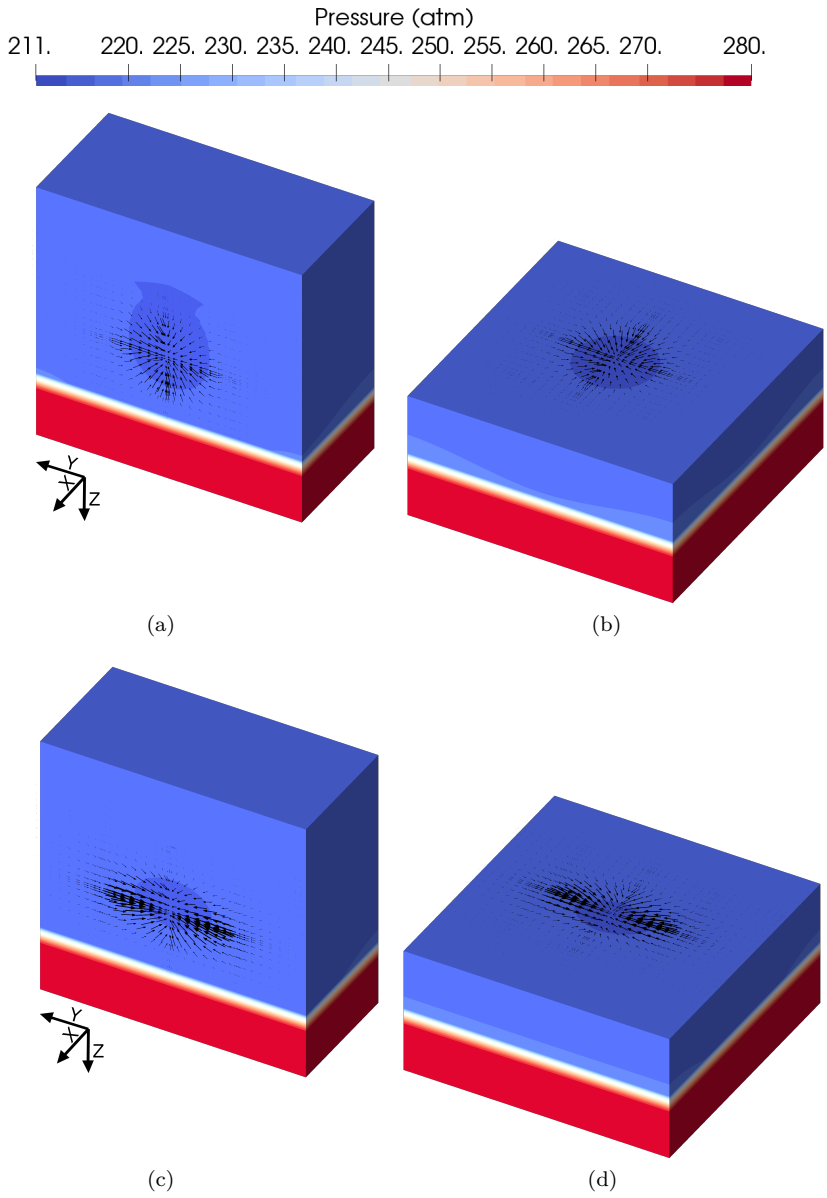


Fig. 10: Pressure field and Darcy velocity in two sections of the computational domain. For the isotropic permeability: a) YZ cross section, and b) XY cross section. For the anisotropic permeability: c) YZ cross section, and d) XY cross section.

that the Engquist-Osher monotone scheme flux is well-suited for the one-phase flow problem. This selection allows deducing an analytical expression of the numerical convective flux. Moreover, we can introduce the numerical convective term in Newton's solver since we can analytically compute its derivatives.

Note that other monotone scheme fluxes could be used for the convective term to ensure the stability of the formulation, such as the Godunov or the Lax-Friedrichs fluxes. However, these fluxes contain non-differentiable expressions such as minima, maxima or suprema. Therefore, it is not straightforward to incorporate these fluxes in Newton's method.

The computational cost of the HDG method is usually larger than other non-mixed methods. The main reason is that the HDG method introduces the diffusive flux and the solution trace as new unknowns. Thus, the elemental contributions of the linear system are larger and their calculation lead to a higher computational cost. Nevertheless, one advantage of the HDG method is that the linear system can be hybridized to obtain a smaller one. The number of unknowns of the hybridized system is of the same order as the number of unknowns of a high-order hybridized CG method [20]. Moreover, the Darcy's velocity is obtained with higher accuracy than non-mixed methods by solving a slightly larger linear system.

In the proposed method, we use a Lagrangian basis of shape functions to generate 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 the used basis. Nevertheless, in our work, we have selected a Lagrangian basis, since we use a non-uniform nodal distribution that approximately minimizes the Lebesgue constant, see [50]. Therefore, the used basis is well-suited for high-order Lagrange interpolation. Moreover, the solution of the problem are the nodal values, which is convenient for practical purposes.

We have shown in the examples the features and advantages of the proposed HDG formulation. Specifically, we have presented numerical evidence of the optimal convergence rates of $P+1$ in L^2-norm, for the pressure, the flux and the Darcy velocity, and also the convergence rate of $P+2$ in L^2-norm for the post-processed pressure. Therefore, the analytical expression for the stabilization parameter ensures the existence and uniqueness of the obtained approximation and the stability and the convergence of formulation. Moreover, we have numerically shown that the mass is conserved at the element level. We have illustrated the capability of the proposed HDG formulation by using structured and unstructured meshes, and heterogeneous anisotropic materials of the reservoir. We have simulated the flow through highly heterogeneous medium using a high-order unstructured mesh adapted to the material properties. Furthermore, we have compared the obtained approximation with an analytical approximation to illustrate the accuracy of the proposed method and to validate our formulation. We also have shown that using high-order methods for both space and time, we achieve high-order accuracy not only for

the scalar variables, but also for the fluxes and velocities with the same rate of convergence even though we use coarse meshes and large time steps.

For the examples presented in this paper, the number of iterations in the Newton-Raphson nonlinear solver is around four if we consider homogeneous materials and two or three more for non-homogeneous materials. The hybridization procedure described in Section 3.5 represents less than 1% of the total time of solving the hybridized system. This is because we use a direct linear solver to solve the linear systems that we obtain with the high-order meshes. Using high-order approximations, the local matrices are larger than for linear approximations. This leads to larger and denser global matrices. Therefore, the computational cost of the global system is higher if direct solvers are used because of the higher number of non-zero entries in the system matrix. For this reason, the cost of the hybridization process is negligible compared to the cost of solving the linear system, even when solving 2D problems. This behavior is in agreement with other authors [25,35,36], where it is stated the low cost of the hybridization procedure.

Several aspects of this work will be analyzed and improved in the near future. First, we will couple the proposed HDG formulation with fully implicit temporal schemes to obtain temporal convergence rates of the same order as the spatial ones, especially for high polynomial degrees. In this way, arbitrary large time steps can be used achieving a high-order accuracy with few stages. Second, to simulate larger 3D domains and more complex models such as black oil and compositional flow models, we will analyze efficient iterative linear solvers and appropriate pre-conditioners. Moreover, we will investigate the cost of the hybridization when compared to an iterative linear solver. Third, we will improve the computational efficiency of our implementation, which is currently programmed in Python, in order to apply our formulation to larger hydrocarbon reservoirs discretized with finer meshes. To this end, we will implement the proposed formulation using a compiled language like Fortran or C++. Moreover, we also consider to parallellize the code to further reduce the wall time. Fourth, we will compare the computational cost and solution accuracy of the HDG method with other state-of-the-art methods.

The datasets generated during and/or analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.

Acknowledgements

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

References

1. Abushaikha, A.S., Voskov, D.V., Tchelepi, H.A.: Fully implicit mixed-hybrid finite-element discretization for general purpose subsurface reservoir simulation. Journal of Computational Physics **346**, 514–538 (2017)
2. Arbogast, T., Juntunen, M., Pool, J., Wheeler, M.F.: A discontinuous Galerkin method for two-phase flow in a porous medium enforcing H (div) velocityand continuous capillary pressure. Computational Geosciences **17**(6), 1055–1078 (2013)
3. Aziz, K., Settari, A.: Petroleum reservoir simulation. Chapman & Hall (1979)
4. Babuska, I., Szabo, B.A., Katz, I.N.: The p-version of the finite element method. SIAM Journal on Numerical Analysis **18**(3), 515–545 (1981)
5. Bastian, P.: A fully-coupled discontinuous Galerkin method for two-phase flow in porous media with discontinuous capillary pressure. Computational Geosciences **18**(5), 779–796 (2014)
6. Brezzi, F., Hughes, T.J., Marini, L.D., Masud, A.: Mixed discontinuous Galerkin methods for Darcy flow. Journal of Scientific Computing **22**(1-3), 119–145 (2005)
7. Butcher, J.C.: Numerical methods for ordinary differential equations. John Wiley & Sons (2016)
8. Chen, Z., Huan, G., Ma, Y.: Computational methods for multiphase flows in porous media. SIAM (2006)
9. Cossé, R.: Oil and Gas Field Development Techniques: Basics of Reservoir Engineering. Éditions Technip (1993)
10. Costa-Solé, A., Ruiz-Gironés, E., Sarrate, J.: An HDG formulation for incompressible and immiscible two-phase porous media flow problems. International Journal of Computational Fluid Dynamics **33**(4), 137–148 (2019)
11. Donea, J., Huerta, A.: Finite element methods for flow problems. John Wiley & Sons (2003)
12. Epshteyn, Y.Y.: Hp primal discontinuous Galerkin finite element methods for two-phase flow in porous media. Ph.D. thesis, University of Pittsburgh (2007)
13. Ern, A., Mozolevski, I., Schuh, L.: Discontinuous Galerkin approximation of two-phase flows in heterogeneous porous media with discontinuous capillary pressures. Computer methods in applied mechanics and engineering **199**(23-24), 1491–1501 (2010)
14. Fabien, M.S., Knepley, M.G., Rivière, B.M.: A hybridizable discontinuous Galerkin method for two-phase flow in heterogeneous porous media. International Journal for Numerical Methods in Engineering **116**(3), 161–177 (2018)
15. Fabien, M.S., Knepley, M.G., Rivière, B.M.: A high order hybridizable discontinuous Galerkin method for incompressible miscible displacement in heterogeneous media. Results in Applied Mathematics p. 100089 (2020)
16. Gargallo-Peiró, A., Roca, X., Peraire, J., Sarrate, J.: Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. International Journal for Numerical Methods in Engineering **103**(5), 342–363 (2015)
17. Gargallo-Peiró, A., Roca, X., Peraire, J., Sarrate, J.: A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization. International Journal for Numerical Methods in Engineering **106**(13), 1100–1130 (2016). Nme.5162
18. Giorgiani, G., Fernández-Méndez, S., Huerta, A.: Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. International Journal for Numerical Methods in Fluids **72**(12), 1244–1262 (2013)
19. Hou, J., Chen, J., Sun, S., Chen, Z.: Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media. Journal of Computational and Applied Mathematics **307**, 262–283 (2016)
20. Huerta, A., Angeloski, A., Roca, X., Peraire, J.: Efficiency of high-order elements for continuous and discontinuous Galerkin methods. International Journal for Numerical Methods in Engineering **96**(9), 529–560 (2013)
21. Hughes, T.J., Masud, A., Wan, J.: A stabilized mixed discontinuous Galerkin method for Darcy flow. Computer Methods in Applied Mechanics and Engineering **195**(25-28), 3347–3381 (2006)

22. Jamei, M., Ghafouri, H.: A novel discontinuous Galerkin model for two-phase flow in porous media using an improved IMPES method. International Journal of Numerical Methods for Heat & Fluid Flow **26**(1), 284–306 (2016)
23. Kennedy, C.A., Carpenter, M.H.: Diagonally implicit Runge-Kutta methods for ordinary differential equations a review. Tech. Rep. NASA/TM-2016-219173, NASA (2016)
24. Ketcheson, D., Seibold, B., Shirokoff, D., Zhou, D.: DIRK schemes with high weak stage order. arXiv preprint arXiv:1811.01285 (2018)
25. Kirby, R., Sherwin, S., Cockburn, B.: To CG or to HDG: a comparative study. Journal of Scientific Computing **51**(1), 183–212 (2012)
26. Klieber, W., Rivière, B.: Adaptive simulations of two-phase flow by discontinuous Galerkin methods. Computer Methods in Applied Mechanics and Engineering **196**(1-3), 404–419 (2006)
27. LeVeque, R.J.: Numerical methods for conservation laws. Springer Science & Business Media (1992)
28. Li, J., Riviere, B.: High order discontinuous Galerkin method for simulating miscible flooding in porous media. Computational Geosciences **19**(6), 1251–1268 (2015)
29. Löhner, R.: Error and work estimates for high-order elements. International Journal for Numerical Methods in Fluids **67**(12), 2184–2188 (2011)
30. Löhner, R.: Improved error and work estimates for high-order elements. International Journal for Numerical Methods in Fluids **72**(11), 1207–1218 (2013)
31. Luo, Y., Feng, M., Xu, Y.: A stabilized mixed discontinuous Galerkin method for the incompressible miscible displacement problem. Boundary Value Problems **2011**(1), 48 (2011)
32. Masud, A., Hughes, T.J.: A stabilized mixed finite element method for Darcy flow. Computer Methods in Applied Mechanics and Engineering **191**(39-40), 4341–4370 (2002)
33. Montlaur, A., Fernandez-Mendez, S., Huerta, A.: High-order implicit time integration for unsteady incompressible flows. International journal for numerical methods in fluids **70**(5), 603–626 (2012)
34. Natvig, J.R., Lie, K.A.: Fast computation of multiphase flow in porous media by implicit discontinuous Galerkin schemes with optimal ordering of elements. Journal of Computational physics **227**(24), 10108–10124 (2008)
35. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. Journal of Computational Physics **228**(9), 3232–3254 (2009)
36. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. Journal of Computational Physics **228**(23), 8841–8855 (2009)
37. Nguyen, N.C., Peraire, J., Cockburn, B.: High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. Journal of Computational Physics **230**(10), 3695–3718 (2011)
38. Pazner, W., Persson, P.O.: Stage-parallel fully implicit Runge-Kutta solvers for discontinuous Galerkin fluid simulations. Journal of Computational Physics **335**, 700–717 (2017)
39. Radu, F.A., Nordbotten, J.M., Pop, I.S., Kumar, K.: A robust linearization scheme for finite volume based discretizations for simulation of two-phase flow in porous media. Journal of Computational and Applied Mathematics **289**, 134–141 (2015)
40. Rivière, B., Wheeler, M.F.: Discontinuous Galerkin methods for flow and transport problems in porous media. Communications in numerical methods in engineering **18**(1), 63–68 (2002)
41. Roca, X., Ruiz-Gironés, E., Sarrate, J.: ez4u: mesh generation environment (2010)
42. Ruiz-Gironés E.; Roca, X., Sarrate, J.: High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation. Computer-Aided Design **72**, 52 – 64 (2016)
43. Russell, T.F., Wheeler, M.F.: Finite element and finite difference methods for continuous flows in porous media. In: The mathematics of reservoir simulation, pp. 35–106. SIAM (1983)
44. Salinas, P., Pavlidis, D., Xie, Z., Jacquemyn, C., Melnikova, Y., Jackson, M.D., Pain, C.C.: Improving the robustness of the control volume finite element method with application to multiphase porous media flow. International Journal for Numerical Methods in Fluids **85**(4), 235–246 (2017)

45. Salinas, P., Pavlidis, D., Xie, Z., Osman, H., Pain, C.C., Jackson, M.D.: A discontinuous control volume finite element method for multi-phase flow in heterogeneous porous media. Journal of Computational Physics **352**, 602–614 (2018)
46. Selley, R.C., Sonnenberg, S.A.: Elements of petroleum geology. Academic Press (2014)
47. Sevilla, R., Huerta, A.: Tutorial on hybridizable discontinuous galerkin (HDG) for second-order elliptic problems. In: Advanced Finite Element Technologies, pp. 105–129. Springer (2016)
48. Versteeg, H.K., Malalasekera, W.: An introduction to computational fluid dynamics: the finite volume method. Pearson education (2007)
49. Wang, Z., Fidkowski Kand Abgrall, R., Bassi, F., Caraeni, D., Cary, A., D, H., Hartmann, R., Hillewaert, K., Huynh, H., et al.: High-order CFD methods: current status and perspective. International Journal for Numerical Methods in Fluids **72**(8), 811–845 (2013)
50. Warburton, T.: An explicit construction of interpolation nodes on the simplex. Journal of engineering mathematics **56**(3), 247–262 (2006)
51. Zhu, J., Taylor, Z., Zienkiewicz, O.: The finite element method: its basis and fundamentals. Elsevier (2013)

A Jacobian terms

In this Appendix, we write the Jacobian terms required to solve Equation (25) of the Newton-Raphson method. First, we deduce the partial derivatives of the numerical convective flux, $\hat{\mathbf{F}}_h \cdot \mathbf{n}$, Equation (12), respect to the pressure, p_h, and its trace, $\hat{p}_h$

$$\frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial p_j} = \begin{cases} \left(\dfrac{\mathbf{F}(p_h) \cdot \mathbf{n} - \hat{\mathbf{F}}_h \cdot \mathbf{n}}{(p_h - \hat{p}_h)} \right) N_j & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} \geq 0, \\ 0 & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} < 0, \end{cases} \tag{36}$$

$$\frac{\partial \hat{\mathbf{F}}_h \cdot \mathbf{n}}{\partial \hat{p}_j} = \begin{cases} \left(\dfrac{-\mathbf{F}(\hat{p}_h) \cdot \mathbf{n} + \hat{\mathbf{F}}_h \cdot \mathbf{n}}{(p_h - \hat{p}_h)} \right) N_j^f & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} \geq 0, \\ \mathbf{F}'(\hat{p}_h) N_j^f & \text{if } (\mathbf{Kg}) \cdot \mathbf{n} < 0, \end{cases} \tag{37}$$

where p_h and $\hat{p}_h$ are defined in Equations (16) and (17), respectively. Note that, for $p_h = \hat{p}_h$ the first equations of Equation (36) and (37) becomes

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

The partial derivatives of Equations (16) to (18) are

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

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

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

Thus, the elemental Jacobian coefficients are

$$
\begin{aligned}
(\mathbf{J}^e_{qq})_{ij,kl} &= \frac{\partial}{\partial q_{kl}}\left(\mathbf{A}^{-1}\mathbf{q}_h, N_i\mathbf{e}_j\right)_e = (\mathbf{A}^{-1}N_k\mathbf{e}_l, N_i\mathbf{e}_j)_e \\
(\mathbf{J}^e_{q\dot{p}})_{ij,k} &= \frac{\partial}{\partial \dot{p}_k}\left((p_i, \nabla\cdot N_i\mathbf{e}_j)_e + (\mathbf{A}^{-1}\mathbf{q}_h, N_i\mathbf{e}_j)_e\right) \\
&= \Delta t a_{ii}\left(\left(-\mathbf{q}_h\mu\mathbf{K}^{-1}\frac{\rho_{ref}c_f N_k}{\rho(p_i)^2}, N_i\mathbf{e}_j\right)_e - \left(N_k, \frac{\partial N_i}{\partial x_j}\right)_e\right) \\
(\mathbf{J}^{e,f}_{q\hat{p}})_{ij,k} &= \frac{\partial}{\partial \hat{p}_k}\langle \hat{p}_h, N_i\mathbf{e}_j\cdot\mathbf{n}\rangle_{\partial e} = \langle N^f_k, N_i\mathbf{n}_j\rangle_{\partial e}
\end{aligned}
$$

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

B Temporal discretization schemes

This second part of the appendix contains the Butcher's tables of the temporal schemes used in the examples. The stability conditions of each one are

Scheme	Stability
Backward	L-stable
DIRK2s2	A-stable
DIRK3s3	L-stable
DIRK4s6	A-stable

see [7, 33, 23, 24] for more details.

Table 6: Butcher's table for the backward scheme.

1	1
	1

Table 7: Butcher's table for the DIRK2-s2 scheme.

$1+\frac{\sqrt{2}}{2}$	$1+\frac{\sqrt{2}}{2}$	
1	$\frac{\sqrt{2}}{2}$	$1+\frac{\sqrt{2}}{2}$
	$\frac{\sqrt{2}}{2}$	$1+\frac{\sqrt{2}}{2}$

Table 8: Butcher's table for the DIRK3-s3 scheme.

γ	γ			$\gamma = 0.4358665215$
$\frac{1+\gamma}{2}$	$\frac{1-\gamma}{2}$	γ		
1	$\frac{-6\gamma^2+16\gamma-1}{4}$	$\frac{6\gamma^2-20\gamma-1}{4}$	γ	
	$\frac{-6\gamma^2+16\gamma-1}{4}$	$\frac{6\gamma^2-20\gamma-1}{4}$	γ	

Table 9: Butcher's table for the DIRK4-s6 scheme.

0.079672377876931	0.079672377876931					
0.464364648310935	0.328355391763968	0.136009256546967				
1.348559241946724	-0.650772774016417	1.742859063495349	0.256472952467792			
1.312664210308764	-0.714580550967259	1.793745752775934	-0.078254785672497	0.311753794172585		
0.989469293495897	-1.120092779092918	1.983452339867353	3.117393885836001	-3.761930177913743	0.770646024799205	
1	0.214823667785537	0.536367363903245	0.154488125726409	-0.217748592703941	0.072226422925896	0.239843012362853
	0.214823667785537	0.536367363903245	0.154488125726409	-0.217748592703941	0.072226422925896	0.239843012362853