Accepted Manuscript

Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media

Jiangyong Hou, Jie Chen, Shuyu Sun, Zhangxin Chen

PII: S0377-0427(16)30030-9
DOI: http://dx.doi.org/10.1016/j.cam.2016.01.050
Reference: CAM 10487

To appear in: Journal of Computational and Applied Mathematics

Received date: 1 July 2015
Revised date: 26 January 2016

Please cite this article as: J. Hou, J. Chen, S. Sun, Z. Chen, Adaptive mixed-hybrid and penalty discontinuous Galerkin method for two-phase flow in heterogeneous media, Journal of Computational and Applied Mathematics (2016), http://dx.doi.org/10.1016/j.cam.2016.01.050

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.
Adaptive Mixed-Hybrid and Penalty Discontinuous Galerkin method for two-phase flow in heterogeneous media

Jiangyong Hou\textsuperscript{1,2}, Jie Chen\textsuperscript{1,2}, Shuyu Sun\textsuperscript{2,3}, and Zhangxin Chen\textsuperscript{1,2,4}

\textsuperscript{1}School of Mathematics and Statistics, Xi'an Jiaotong University, Shaanxi, P.R.China,710049
\textsuperscript{2}Center for Computational Geosciences, Xi'an Jiaotong University, Shaanxi, P.R.China,710049
\textsuperscript{3}Division of Physical Sciences and Engineering (PSE), King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
\textsuperscript{4}Schulich School of Engineering, University of Calgary, 2500 University Drive NW, Calgary, AB, Canada T2N 1N4

Abstract

In this paper, we present a hybrid method, which consists of a mixed-hybrid finite element method and a penalty discontinuous Galerkin method, for the approximation of a fractional flow formulation of a two-phase flow problem in heterogeneous media with discontinuous capillary pressure. The fractional flow formulation is comprised of a wetting phase pressure equation and a wetting phase saturation equation which are coupled through a total velocity and the saturation affected coefficients. For the wetting phase pressure equation, the continuous mixed-hybrid finite element method space can be utilized due to a fundamental property that the wetting phase pressure is continuous. While it can reduce the computational cost by using less degrees of freedom and avoiding the post-processing of velocity reconstruction, this method can also keep several good properties of the discontinuous Galerkin method, which are important to the fractional flow formulation, such as the local mass balance, continuous normal flux and capability of handling the discontinuous capillary pressure. For the wetting phase saturation equation, the penalty discontinuous Galerkin method is utilized due to its capability of handling the discontinuous jump of the wetting phase saturation. Furthermore, an adaptive algorithm for the hybrid method together with the centroidal Voronoi Delaunay triangulation technique is proposed. Five numerical examples are presented to illustrate the features of proposed numerical method, such as the optimal convergence order, the accurate and efficient velocity approximation, and the applicability to the simulation of water flooding in oil field and the oil-trapping or barrier effect phenomena.

Keywords. Brezzi-Douglas-Marini, Raviart-Thomas, mixed-hybrid, continuity of wetting phase pressure, penalty discontinuous Galerkin, discontinuous nonlinear interface condition, centroidal Voronoi Delaunay triangulation

§ 1 Introduction

In the article we focus on the numerical methods for the model of fractional flow formulation of a two-phase flow problem. The fractional flow formulation has two equations, the pressure equation and the saturation equation which are coupled through the total velocity and the saturation affected coefficients. Depending on different combination of the basic equations of the two-phase flow

\textsuperscript{*}The work is supported by the National Natural Science Foundation of China (No.11371288 and No.11401467), China Postdoctoral Science Foundation (No. 2013M542334. and No. 2015T81012), Natural Science Foundation of Shaanxi Province (No. 2015JQ1012) and the Fundamental Research Funds for the Central Universities.

\textsuperscript{†}Corresponding to: Jie Chen, School of Mathematics and Statistics, Xi'an Jiaotong University, Shaanxi, P.R.China,710049 (chenjiexjtu@mail.xjtu.edu.cn)
problem, there are several possibilities for the unknowns of pressure and saturation in the possible fractional flow formulation [1]. For instance, in [2], the global pressure and the nonwetting phase saturation are utilized as the unknowns for the equation system. Both of these two variables are discontinuous in the heterogeneous media. In their paper, the penalty discontinuous Galerkin (PDG) methods are used to tackle the discontinuous interface conditions of the unknowns. They also provide an accurate but complicated post-processing technique to recover the total velocity from the pressure equation. In [3], the wetting phase pressure and the wetting phase saturation are used as the unknown variables. The wetting phase saturation is discontinuous but the wetting phase pressure is continuous. In their paper, they also use the PDG methods to solve the problem and provide a simple velocity reconstruction technique. But this reconstruction is not accurate enough to obtain a locally balanced velocity then impairs the convergence order of saturation and pressure.

In this paper, for the model with variables of wetting phase pressure and wetting phase saturation, we develop a new hybrid method based on the property that the wetting phase pressure is always continuous. The hybrid method is composed of a continuous mixed-hybrid finite element (MHFE) method and the PDG method, which are utilized for the wetting phase pressure equation and the wetting phase saturation equation respectively.

For the wetting phase pressure equation, the wetting phase pressure is always continuous because the wetting phase (such as water) is assumed to be everywhere in a reservoir. Due to this continuity property, the continuous MHFE method, which requires a weak continuity in the pressure across the edges, can be used to solve the wetting phase pressure equation. While it can reduce the computational cost, the MHFE method can also keep several good properties of the discontinuous Galerkin (DG) method which are important to the fractional flow formulation. For reducing the computational cost, the velocity is computed directly and no extra post-processing of velocity reconstruction is needed in this method. Furthermore, continuous MHFE method uses less degrees of freedom than the PDG method. The properties which the MHFE method can keep include the local mass conservation, the continuous normal flux and the capability of tackling with the discontinuity of the capillary pressure. The first property is a fundamental requirement for the porous medium flow and is marched by the MHFE method itself. The second property is also a basic assumption for the flow in porous medium and is satisfied by the MHFE method with specific finite element spaces such as Raviart-Thomas (RT) finite element and Brezzi-Douglas-Marini (BDM) finite element spaces which can provide accurate velocity with continuous normal flux [4]. For the third property, the MHFE method can deal with the discontinuous capillary pressure term due to its flexible hybrid scheme [5].

For the wetting phase saturation equation, the wetting phase saturation is discontinuous in the heterogeneous media and satisfies a nonlinear interface condition on the interface between two different permeable rocks. Following [2], we use the PDG method to solve the saturation equation, since it can easily handle the discontinuous jump of saturation on the interface. Thus the combination of MHFE and PDG methods is very rational and fully takes the advantage of the continuity property of the wetting phase pressure in the model. The analysis of the combined mixed finite element method and the penalty discontinuous Galerking method can be found in [6]. Furthermore, we provide an adaptive strategy with centroidal Voronoi Delaunay triangulation (CVDT) technique [7–10] for our numerical method. This adaptive algorithm produces conforming adaptive meshes on which the original limiters [11] for DG methods can still be used. By contrast, in [3], a hp adaptive algorithm for the PDG method was performed and the resulting mesh is non-matching on which the original limiters cannot be directly used. So they need to provide some remedies to the original slope limiters on the mesh having hanging nodes. Our adaptive algorithm is a preliminary result and will be promoted in the future work.

There are some other combination methods for the two phase flow problem in the literatures [5, 12–14]. For example, in [12], the mean pressure was used in the model of fractional flow formulation of two-phase flow problem, and the saturation equation was split into diffusion part and convection
part which were separately solved by the MHFE and the DG method respectively. In [5,13], instead of the fractional flow formulation another formulation of a two-phase flow problem was developed, and the combination of MHFE and DG method was used to solve it. In [14], the mixed finite element method was combined with a finite volume method to solve the two phase flow problem, but they didn’t consider the discontinuities of the capillary pressure.

The rest of the paper is organized as follows: In addition to the introduction and conclusions, this paper is composed of three parts. Section 2 is the first part and consists of two subsections; we introduce the governing equations of the fractional flow formulation of a two-phase flow problem and the corresponding interface conditions in Subsections 2.1 and 2.2, respectively. The second part, Section 3, comes in four subsections. In Subsections 3.1 and 3.2, the MHFE method and the PDG method are used for the pressure equation and the saturation equation, respectively. In Subsection 3.3, the sequential algorithms of the MHFE-PDG method and the PDG-PDG method are presented. The adaptive strategy is described in Subsection 3.4. In the last section, five numerical examples are provided to illustrate the effectiveness of the proposed numerical method.

§ 2 Problem model

2.1 Mathematical formulation We consider two immiscible incompressible fluids in porous media and there is no mass transfer between the phases. Various model equations for the two-phase flow problem can be found in [1]. Here we consider the phase formulation whose unknown variables are wetting phase pressure, total velocity and wetting phase saturation (p_w, u_t and S_w), and in the absence of gravity the equations are

\[
\begin{align*}
\mathbf{u}_t &= -\lambda_w \nabla p_w - \lambda_n D \nabla p_c, \\
\nabla \cdot \mathbf{u}_t &= q_n + q_n, \\
\theta \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda_n f_w \nabla S_w) + \nabla \cdot (f_w \mathbf{u}_t) &= q_w,
\end{align*}
\]

where \( D \) is the absolute permeability tensor; \( \theta \) denotes the porosity of the medium; \( p_c \) is the capillary pressure; \( \lambda_n \) and \( \lambda_t \) are nonwetting and total mobilities, respectively; \( f_w \) is the fractional flow function; \( q_n \) and \( q_w \) are sink/source term.

Throughout this paper we use the Brooks-Corey model [15] for the coefficients such as the capillary pressure \( p_c \), in which some of these coefficients are non-linear functions and defined as follows:

\[
\begin{align*}
p_c(S_w) &= p_d S_w^{-\frac{1}{2}} \frac{dp_c}{dS_w}(S_w) = -\frac{p_d}{\theta(1 - S_{rn} - S_{rw})} S_w^{-\frac{1}{2}+\theta}, \\
\lambda_n(S_w) &= \frac{(1 - S_w)^2(1 - S_w^{2+\theta})}{\mu_n}, \\
\lambda_w(S_w) &= \frac{S_w^{2+\theta}}{\mu_w}, \\
\lambda_t &= \lambda_n + \lambda_w, \quad f_w = \frac{\lambda_w}{\lambda_t},
\end{align*}
\]

where \( p_d \) is a threshold pressure; \( \lambda_w \) is the wetting phase mobility; \( \mu_n \) and \( \mu_w \) are the non-wetting and wetting phase viscosities; \( \theta \) is the parameter associated with pore size distribution; \( S_{rn} \) is the effective saturation defined as \( (S_w - S_{rn})/(1 - S_{rn} - S_{rw}) \), \( S_{rw} \) and \( S_{rn} \) are residual wetting phase saturation and residual nonwetting phase saturation respectively. The value of the wetting phase saturation \( S_w \) lies in the interval \( [S_{rn}, 1 - S_{rn}] \subseteq [0, 1] \).

Some notations for the mesh are given below: \( \Omega \) is the domain; \( \partial \Omega \) is the boundary of the domain; \( T_h \) is the partition of \( \Omega \); \( K \) is the finite element in \( T_h \); \( \partial K \) is the edges of element \( K \); \( E_h := \{ e: \text{all edges in } T_h \} \), is the set of all edges contained in \( T_h \); \( \mathcal{E}_h^i \) is the set of all interior edges contained in \( T_h \).
We consider two types of boundary conditions for equations (2.1)-(2.3), one is the Robin-Neumann boundary condition:

\[
(S_w u_t + \lambda_n f_w D \nabla p_c) \cdot n = S_{in} u_t \cdot n, \quad \text{on } \Gamma_{sM},
\]

\[
\lambda_n f_w D \nabla p_c \cdot n = \tilde{g}_N, \quad \text{on } \Gamma_{sN},
\]

\[
p_w = p_{\text{dir}}, \quad \text{on } \Gamma_{pD},
\]

\[
u_t \cdot n = g_N, \quad \text{on } \Gamma_{pN},
\]

where \(S_{in}\) is a known value for the wetting phase saturation at the inflow boundary. The other one is Neumann-Drichlet boundary condition:

\[
u_w \cdot n = \tilde{g}_N, \quad \text{on } \Gamma_{sN},
\]

\[
S_w = S_{\text{dir}}, \quad \text{on } \Gamma_{sD},
\]

\[
p_w = p_{\text{dir}}, \quad \text{on } \Gamma_{pD},
\]

\[
u_t \cdot n = g_N, \quad \text{on } \Gamma_{pN},
\]

where \(u_w\) is the wetting phase velocity defined as \(u_w = \lambda_n f_w D \nabla p_c + f_w u_t\). The latter one will be used in the oil trapped experiment later. The whole boundary of the porous medium domain \(\partial \Omega\) is divided into three mutually disjoint parts: the inflow, noflow, and outflow boundaries (\(\Gamma_{in}, \Gamma_{no}, \Gamma_{out}\)), respectively. In addition, according to the boundary conditions for the pressure equation (or saturation equation), \(\Gamma_{pN}, \Gamma_{pD}\) and \(\Gamma_{pM}\) (or \(\Gamma_{sN}, \Gamma_{sD}\) and \(\Gamma_{sM}\)) represent the Neumann, Dirichlet and mixed boundaries respectively. The correspondence between \(\Gamma_{pN}, \Gamma_{pD}, \Gamma_{pM}\) (or \(\Gamma_{sN}, \Gamma_{sD}, \Gamma_{sM}\)) and \(\Gamma_{in}, \Gamma_{no}, \Gamma_{out}\) will be given in the specific numerical examples below.

### 2.2 Interface conditions

In order to test the barrier effect phenomenon of two phase flow, the nonlinear interface condition discussed in [2, 12] will be introduced here. Following [13], we assume an initially fully water saturated domain \((\Omega = \Omega_I \cup \Omega_{II})\) with an interface \(\Gamma_J\) between two different sands, and the oil is injected from the inflow part of boundary \(\Gamma_{in}\), see Figure 1. For convenience, we also assume that \(\Omega_I\) stands the coarse sand and \(\Omega_{II}\) is the fine sand. The opposite case is also available that \(\Omega_I\) is finer than \(\Omega_{II}\), which is the case 3b in the third numerical experiment below.

![Diagram](image)

Figure 1: The interface (dashed line) between two subdomains with different rock properties

The process of the phenomenon is described briefly below. First, oil approaches the material interface but can not penetrate it and begin to accumulate. In this case, only water pressure \(p_w\) is continuous on the interface, capillary pressure \(p_c\) and saturation \(S_w\) are discontinuous and satisfy:

\[
\begin{aligned}
  p_c|_{\Omega_I} &= p_d|_{\Omega_I} \tilde{S}^w, \\
  p_c|_{\Omega_{II}} &= p_d|_{\Omega_{II}}, \\
  S^I_w - S^{II}_w &= S^I_w - (1 - S^{II}_w).
\end{aligned}
\]

(2.16)

Then, when more and more oils accumulate at the interface and the capillary pressure on the coarse side exceeds the entry pressure of the other side \((p_c|_{\Omega_I} \geq p_d|_{\Omega_{II}})\), the oils begin to penetrate and enter...
the fine sand. At this time, both $p_w$ and $p_c$ are continuous, but saturation $S_w$ is still discontinuous and satisfies:

$$S^I_w - S^II_w = S_w^I - (1 - S^II_{rw} - S^II_{rn}) \left( \frac{p^I_d}{p^II_d} \right)^{\theta_{II}} \left( \frac{S^I_w - S^II_{rn}}{1 - S^II_{rw} - S^II_{rn}} \right)^{\theta_{II}} = S^II_{rw}. \quad (2.17)$$

We note that a critical point of saturation can be found when the capillary pressure on coarse side increases to the value equivalent to the threshold pressure on fine side. That is, deducing from $p_c|_{\Omega_I} = p^II_d$, we can define the critical point $S^*_w$, which is used to judge whether the nonwetting phase can penetrate the material interface, as follows,

$$S^*_w := (1 - S^I_{rw} - S^I_{rn}) \left( \frac{p^I_d}{p^II_d} \right)^{\theta_{II}} + S^I_{rw}. \quad (2.18)$$

So the interface conditions can be rewritten in the form below. For capillary pressure,

$$p_c|_{\Omega_I} - p_c|_{\Omega_{II}} = \begin{cases} p^I_d S^w w^{\theta_{II}} - p^II_d, & S^I_w > S^*_w, \\ 0, & S^I_w \leq S^*_w, \end{cases} \quad (2.19)$$

and for wetting phase saturation,

$$J(S_w) := S^I_w - S^II_w = \begin{cases} S^I_w - (1 - S^II_{rn}), & S^I_w > S^*_w, \\ S^I_w - (1 - S^II_{rn} - S^II_{II}) \left( \frac{p^II_d}{p^I_d} \right)^{\theta_{II}} \left( \frac{S^I_w - S^II_{rn}}{1 - S^II_{rw} - S^II_{rn}} \right)^{\theta_{II}} - S^II_{rw}, & S^I_w \leq S^*_w. \end{cases} \quad (2.20)$$

Condition (2.20) is the same with that described in [2] except that the wetting phase (instead of the nonwetting phase) is used as the saturation variable. Moreover, (2.19) is only written for the capillary pressure and not for the wetting phase pressure, since the variable $p_w$ is always continuous in the problem discussed. Noting that if the sub-domain $\Omega_I$ has a finer texture than $\Omega_{II}$, all the relationship above can be treated in a similar manner with superscript $I$ and $II$ reversed.

§ 3 Discrete Schemes

3.1 Pressure and Velocity Approximation In this section, we use the mixed-hybrid finite element method with $H(div)$ finite element spaces [4] to discretize the pressure and velocity equations.

At first, we introduce two classical $H(div)$ finite element spaces, the Raviart-Thomas($RT$) finite element spaces and the Brezzi-Douglas-Marini($BDM$) finite element spaces [4]. On triangles, the $RT$ spaces are defined for each $r \geq 0$ by

$$V_h(K) = \left( P_r(K)^2 \right) \oplus \left( (x_1, x_2) P_r(K) \right), \quad W_h(K) = P_r(K), \quad (3.1)$$

where the notation $\oplus$ indicates a direct sum and $(x_1, x_2) P_r(K) = \{x_1 P_r(K), x_2 P_r(K)\}$. The $BDM$ spaces on triangles are defined for each $r \geq 1$ by

$$V_h(K) = P_r(K)^2, \quad W_h(K) = P_{r-1}(K). \quad (3.2)$$

The $BDM$ spaces on triangles lie between corresponding $RT$ spaces, are of smaller dimension than the $RT$ space of the same index, and provide asymptotic error estimates for the vector variable of the same order as the corresponding $RT$ space [16]. We will compare the $BDM_1$ space and the $RT_1$ space in the numerical examples.

Upon the mesh partitions and $H(div)$ spaces given above, we define the finite element spaces:

$$V_h = \{ v \in H(div, \Omega) : v|_K \in V_h(K), \forall K \in T_h \}, \quad (3.3)$$

$$W_h = \{ w \in L^2(\Omega) : w|_K \in W_h(K), \forall K \in T_h \}. \quad (3.4)$$
And we define the norms:
\[
\|w\|_0 := \|v\|_{L^2(\Omega)}, \quad \forall w \in W_h, \quad (3.5)
\]
\[
\|v\|_0 := \left( \sum_{i=1}^d \|v_i\|_{L^2(\Omega)}^2 \right)^{1/2}, \quad \forall v = \{v_i\}_{i=1}^d \in V_0, \quad d = 2 \text{ or } 3, \quad (3.6)
\]
\[
\|v\|_{div} := \left( \|v\|^2 + \|\nabla \cdot v\|^2 \right)^{1/2}, \quad \forall v \in V_h, \quad (3.7)
\]
\[
\|v\|_{div} := \|\nabla \cdot v\|_0, \quad \forall v \in V_h, \quad (3.8)
\]
The normal components of the functions in \(V_h\) are continuous across the interior boundaries in \(E^i_h\).

For the stability condition, both the RT spaces (3.1) and the BDM spaces(3.2) satisfy the discrete inf-sup condition [4,16]
\[
\sup_{0 \neq w \in V_h} \frac{|(\nabla \cdot v, w)|}{\|v\|_{div}} \quad \forall w \in W_h, \quad (3.10)
\]

The mixed finite element schemes have an equivalent hybrid formulation, which relaxes the continuity constraints of the flux variable across element interfaces and imposes them instead by introducing Lagrange multipliers and requiring additional variational equations [4,16–18]. Relaxing the continuity constraint on \(V_h\) and defining the space of Lagrange multipliers as follows,
\[
\tilde{V}_h = \{ v \in L^2(\Omega)^d : v|_K \in V_0(K), \quad \forall K \in T_h \}, \quad d = 2 \text{ or } 3, \quad (3.11)
\]
\[
L_h = \left\{ \eta \in L^2 \left( \bigcup_{e \in E^i_h \setminus \Gamma_{in}} e \right) : \eta|_e \in V_h \cdot \mathbf{n}_e, \quad \forall e \in E^i_h \right\}, \quad (3.12)
\]

where \(\mathbf{n}\) is the unit normal vector on edge \(e\). The functions in the space (3.12) are the Lagrange multipliers to enforce the required continuity on the original \(V_h\) so that the normal components of the functions in \(\tilde{V}_h\) are continuous across the interior boundaries in \(E^i_h\). Actually, these Lagrange multipliers are pressures on the interior edges and impose a weak continuity in the pressure variable [19]. The wetting phase pressure \(p_w\) is always continuous in the discussed problem, so the MHFE method can be used to solve the equation (2.1).

Based on (2.5) and (2.6), the wetting phase mobility \(\lambda_w\) vanishes as \(S_w \rightarrow S_{rw}\), and the non-wetting phase mobility \(\lambda_n\) vanishes as \(S_n \rightarrow S_{rn}\), where \(S_n\) is the nonwetting phase saturation and satisfies \(S_n = 1 - S_{rw}\). In general, the residual saturations are small enough such that \(S_{rw} + S_{rn} \ll 1\). So, the total mobility \(\lambda_t\) is always nonzero, since the value of \(S_w\) and \(S_n\) cannot be \(S_{rw}\) and \(S_{rn}\) simultaneously. Dividing the equation (2.1) by \(\lambda_t D\), multiplying arbitrary test function \(v \in \tilde{V}_h\), and integrating the result over any element \(K \in T_h\), we get,
\[
\int_K \frac{1}{\lambda_t} D^{-1} \mathbf{u}_t \cdot \mathbf{v} = -\int_K \nabla p_w \cdot \mathbf{v} - \int_K f_n \nabla p_c \cdot \mathbf{v}, \quad (3.13)
\]
where \(f_n = \frac{\nabla p_n}{K_t}\). According to the chain rule, the gradient of capillary pressure can be written as
\[
\nabla p_c = \frac{dp_c}{dS_w} \nabla S_w, \quad (3.14)
\]
Substituting (3.14) into (3.13), we obtain from Green’s formula that,
\[
\int_K \frac{1}{\lambda_t} D^{-1} \mathbf{u}_t \cdot \mathbf{v} = \int_K p_w \nabla \cdot \mathbf{v} - \int_{\partial K} p_w \mathbf{v} \cdot \mathbf{n}_K - \int_K f_n \frac{dp_c}{dS_w} \nabla S_w \cdot \mathbf{v}, \quad (3.15)
\]
where \(\mathbf{n}_K\) is the outward unit normal vector at \(\partial K\).

Now, we give the variational form of the equations (2.1)-(2.2) discretized by MHFE method. Supposing that the saturation at the previous time step \(S_{kn}^0\) is known, where the indices \(k\) and \(k+1\)
stand the time step at the previous time \( t_k \) and at the current time \( t_{k+1} \) respectively. Considering
the equation (3.15) over all the elements together with the boundary conditions (2.10)-(2.11) or
(2.14)-(2.15), we can obtain the following variational form,

\[
\sum_{K \in T_h} \int_K \frac{1}{\lambda^k} D_K \mathbf{u}^{k+1} \cdot \mathbf{v}_K - \sum_{e \in \partial K} \int_e \xi^{k+1} \mathbf{n} \cdot \mathbf{v}_K \]

\[
= - \sum_{K \in T_h} \int_K \nabla \cdot \mathbf{u}^{k+1} \mathbf{v}_K - \sum_{e \in \partial K} \int_e p^{n+1} \mathbf{v} \cdot \mathbf{n}_e, \tag{3.16}
\]

\[
\sum_{K \in T_h} \int_K \nabla \cdot \mathbf{u}^{k+1} \mathbf{v}_K = \sum_{K \in T_h} \int_K (g^{k+1} + h^{k+1}) \omega K, \tag{3.17}
\]

\[
\sum_{e \in \partial K} \int_e [\mathbf{u}^{k+1} \cdot \mathbf{n}_e] \eta + \sum_{e \in \partial T} \int_e [\mathbf{u}^{k+1} \cdot \mathbf{n}_e] \eta = \sum_{e \in \partial T} \int e \mathbf{f}^{k+1} \cdot \mathbf{n}_e \eta, \tag{3.18}
\]

where the coefficients \( \lambda^k \) and \( (f_w \frac{dp}{ds})^k \) are \( \lambda^k(S^k_w) \) and \( (f_w \frac{dp}{ds})^k(S^k_w) \) respectively.

At last, we briefly discuss the well-posedness of the scheme of MHFE method. For example, after
the introduction of basis functions in \( \tilde{V}_h \), \( W_h \), and \( L_h \), the variational equations (3.16)- (3.18) can be expressed in the matrix form:

\[
\begin{pmatrix}
A & B & C
\end{pmatrix}
\begin{pmatrix}
U \\
B^T \\
C^T
\end{pmatrix}
= \begin{pmatrix}
U \\
P \\
L
\end{pmatrix} = \begin{pmatrix}
f_1 \\
f_2 \\
f_3
\end{pmatrix} \tag{3.19}
\]

where \((U, P, L)^T\) is the degrees of freedom of \((\mathbf{u}^{k+1}, p^{k+1}, \xi^{k+1})\). The basis functions in \( \tilde{V}_h \) are nonzero only in one element \( K \), due to which the matrix \( A \) has now an elementwise block-diagonal structure with each block corresponding to a single element. Hence \( A \) is easily inverted at the element level. This, together with the first equation in (3.19), leads to

\[
U = A^{-1}BP - A^{-1}CL + A^{-1}f_1. \tag{3.20}
\]

Substituting it into the second and third equations in (3.19), we see that

\[
B^T A^{-1}BP + B^T A^{-1}CL = f_2 + B^T A^{-1}f_1, \tag{3.21}
\]

\[
C^T A^{-1}BP + C^T A^{-1}CL = f_3 + C^T A^{-1}f_1. \tag{3.22}
\]

By (3.10), \( B^T A^{-1}B \) is symmetric and positive definite, so the equation (3.21) yields

\[
P = (B^T A^{-1}B)^{-1}(f_2 + B^T A^{-1}f_1) - (B^T A^{-1}B)^{-1}(B^T A^{-1}CL) \tag{3.23}
\]

Consequently, (3.21) and (3.22) can be further reduced to

\[
ZL = F, \tag{3.24}
\]

where

\[
Z = \left( C^T A^{-1}C - (C^T A^{-1}B) (B^T A^{-1}B)^{-1} (B^T A^{-1}CL) \right), \tag{3.25}
\]

\[
F = -f_3 + C^T A^{-1}f_1 - (C^T A^{-1}B)(B^T A^{-1}B)^{-1}(f_2 + B^T A^{-1}f_1). \tag{3.26}
\]

The matrix \( Z \) is symmetric, positive definite, and sparse with a narrow stencil, for more details about
the existence, uniqueness and convergence analysis refer to [4, 16–18, 20, 21]. We remark that the
number of Lagrange multipliers, which corresponds to the size of the overall global system (3.24),
remains the same for \( RT_1 \) space and \( BDM_1 \) space, since the two additional flux unknowns of \( RT_1 \)
space do not contribute to the normal fluxes across the interelement edges [16].
3.2 Saturation Approximation In this section, we apply the penalty discontinuous Galerkin methods to the saturation equation (2.3). Some notations and the finite element space for DG methods are defined:

\[
\{\psi\} := \frac{1}{2}(\psi^- + \psi^+), \quad [\psi] := \psi^- - \psi^+, \quad \forall e \in E_h,
\]

\[
\{\psi\} := \psi, \quad [\psi] := \psi, \quad \forall e \in \partial \Omega,
\]

\[
X_h := \{\psi \in L^2(\Omega) : \psi|_K \in P_1(K), \forall K \in T_h\},
\]

where \(\psi^\pm\) are the restrictions of \(\psi\) on two adjacent elements \(K^\pm\) respectively. The penalty parameters of DG method are set to \(\epsilon = 1\), \(\sigma_c = 1\), and \(\beta = 1\), which imply the nonsymmetric interior penalty Galerkin (NIPG) method [22].

We introduce another index \(m\) to indicate the nonlinear iteration step. The saturation \(S_w^{k,m}\) means that the solution is at the time \(t_k\) and at the \(m\)-th step of the nonlinear iteration. Supposing that the saturations \(S_w^k(S_w^0\) is the initial condition) and \(S_w^{k,m}\) \((S_w^{0} = S_w^k\) are given, the equation (2.3) equipped with Robin-Neumann boundary conditions (2.8)-(2.11) can be discretized as:

\[
\begin{aligned}
\text{find} \quad S_w^{k+1,m+1} &\in X_h, \text{ such that, for all } \psi \in X_h, \\
\sum_{K \in T_h} \int_K \phi_K S_w^{k+1,m+1}\psi_K &- \sum_{K \in T_h} \int_K (\lambda_n f_w \frac{dp_w}{dS_w})^{k,m} \nabla S_w^{k+1,m+1} \cdot \nabla \psi_K \\
+ \sum_{e \in \partial T_h} \int_{\partial K} (\lambda_n f_w \frac{dp_w}{dS_w})^{k,m} \nabla S_w^{k+1,m+1} \cdot n_e \psi &- \sum_{e \in \partial K} \int_e \phi_e S_w^{k+1,m+1} \psi &= 0, \\
- \varepsilon &\sum_{e \in \partial T_h} \int_{\partial K} (\lambda_n f_w \frac{dp_w}{dS_w})^{k,m} \nabla S_w^{k+1,m+1} \cdot n_e \psi \\
- \sum_{e \in \partial T_h} \int_{\partial K} (f_w^{\uparrow})^{k,m} \psi &= 0,
\end{aligned}
\]

where \((\lambda_n f_w dS_w/dp_w)^{k,m}\) or \(f_w^{k,m}\) means the function \(\lambda_n f_w dS_w/dp_w\) or \(f_w\) takes the value at \(S_w^{k,m}\). The quantity \(f_w^{\uparrow}\) is called the upwind flux which is determined with respect to the normal component of the total velocity \(u_t\), such that for all \(e \in \partial K^- \cap \partial K^+\),

\[
f_w^{\uparrow} = \begin{cases} 
 f_w|_{K^-}, & u_t \cdot n_e \leq 0, \\
 f_w|_{K^+}, & u_t \cdot n_e > 0,
\end{cases}
\]

where the normal vector \(n_e\) points from \(K^+\) to \(K^-\). If \(e\) is on the boundary, \(f_w^{\uparrow} = f_w\). We remark that, in (3.30), the time derivative \(dS_w/dt\) is discretized by the Euler scheme, the diffusion term \(\nabla \cdot (\lambda_n f_w D \frac{dp_w}{dS_w} \nabla S_w)\) is discretized by the NIPG method, and the convection term \(\nabla \cdot (f_w u_t)\) is treated by the upwind scheme.

Similarly, the variational form incorporated with Neumann-Dirichlet boundary conditions (2.12)-(2.15) and interface condition (2.20) reads:

Find \(S_w^{k+1,m+1} \in X_h\), such that, for all \(\psi \in X_h\),

\[
\begin{aligned}
\sum_{K \in T_h} \int_K \phi_K S_w^{k+1,m+1}\psi_K &- \sum_{K \in T_h} \int_K (\lambda_n f_w \frac{dp_w}{dS_w})^{k+1,m} \nabla S_w^{k+1,m+1} \cdot \nabla \psi_K \\
+ \sum_{e \in \partial T_h} \int_{\partial K} (\lambda_n f_w \frac{dp_w}{dS_w})^{k,m} \nabla S_w^{k+1,m+1} \cdot n_e \psi &- \sum_{e \in \partial K} \int_e \phi_e S_w^{k+1,m+1} \psi &= 0, \\
- \varepsilon &\sum_{e \in \partial T_h} \int_{\partial K} (\lambda_n f_w \frac{dp_w}{dS_w})^{k,m} \nabla S_w^{k+1,m+1} \cdot n_e \psi \\
- \sum_{e \in \partial T_h} \int_{\partial K} (f_w^{\uparrow})^{k,m} \psi &= 0,
\end{aligned}
\]
\[
= \sum_{K \in \mathcal{T}_h} \int_K \phi S_w^{k,0} \psi_K - \varepsilon \sum_{e \in \Gamma_J} \int_e \left( (\lambda_w \frac{dp_w}{dS_w})^{k,m} D\nabla \psi \cdot n_e \right) J(S_w^{k,m}) \\
+ \sum_{e \in \Gamma_J} \int_e \frac{d\pi}{|e|} [\psi] \beta_{\pi} \psi + \sum_{e \in \Gamma_J} \int_e \psi_{\nabla}^{k+1} \psi - \sum_{e \in \Gamma_D} \int_e g_{N}^{k+1} \psi \\
+ \sum_{K \in \mathcal{T}_h} \int_K (f_w)^{k,m} u_K^{k+1} \cdot \psi_K - \sum_{e \in \Gamma_D} \int_e \left( (f_w)^{k,m} u_K^{k+1} \cdot n_e \right) [\psi] \\
- \varepsilon \sum_{e \in \Gamma_D} \int_e \left( (\lambda_w \frac{dp_w}{dS_w})^{k,m} D\nabla \psi \cdot n_e \right) S_{\text{dir}}^{k+1} \\
+ \sum_{e \in \Gamma_D} \int_e \frac{\sigma_{\pi}}{|e|} p_{\pi}^{k+1} [\psi]
\]

where \( J(S_w) \) is the interface condition of saturation described in (2.20). Both the interface condition and the Dirichlet boundary condition are imposed weakly by means of the addition of a penalty term in the variational formulation.

The diffusion coefficient of the saturation equation may be degenerated in the heterogeneous media, so in (3.30) and (3.32) we use a Picard iteration to approximate those nonlinear coefficients. That is, at time step \( k + 1 \), \( S_w^{k,m} \) \( (m \geq 0) \), \( S_w^{k+1,0} = S_w^{k} \) and \( u_K^{k+1} \) are given, we can solve the equation (3.30) or (3.32) to get \( S_w^{k+1,m+1} \), then check the error

\[
|| S_w^{k+1,m+1} - S_w^{k,m+1} ||_{L^2(\Omega)} < \epsilon_{\text{tol}},
\]

where \( \epsilon_{\text{tol}} \) is a prescribed tolerance. If the criterion (3.33) is not satisfied, the iteration goes on with the new solution \( S_w^{k,m} = S_w^{k+1,m+1} \), else stop the iteration and take the solution at \( t_{k+1} \) as \( S_w^{k+1} = S_w^{k+1,m} \). The detailed algorithm will be discussed in next section.

### 3.3 Sequential algorithms of MHFE-PDG and PDG-PDG methods

In this section, we give the sequential algorithms of the MHFE-PDG method and the PDG-PDG method. In a sequential approach, the idea is to decouple the equations by time-lagging the coefficients [22]. The abbreviation MHFE-PDG indicates the combination of MHFE method and PDG method. Similarly, PDG-PDG indicates the combination of two penalty DG methods.

Firstly, we briefly recall the simple velocity reconstruction based on the PDG method described in [3]. We rewrite the equations (2.1) and (2.2) into one second-order equation with one unknown \( p_w \), as follows,

\[
- \nabla \cdot (\lambda_w D\nabla p_w + \lambda_e D\nabla p_e) = q_w + q_n.
\]

The variational form of the equation (3.34) together with boundary conditions (2.10)-(2.11) or (2.14)-(2.14) reads as:

\[
\text{find } p_w^{k+1} \in X_h, \text{ for all } \psi \in X_h, \quad \\
\sum_{K \in \mathcal{T}_h} \int_K \lambda_w^{k} D\nabla p_w^{k+1} \psi = \sum_{e \in \Gamma_D} \int_e \lambda_w^{k} D\nabla p_w^{k+1} \cdot n_e \psi + \sum_{e \in \Gamma_D} \int_e \sigma_{\pi}^{k+1} p_{\pi}^{k+1} \psi \\
+ \sum_{e \in \Gamma_D} \int_e \left( (\lambda_w \frac{dp_w}{dS_w})^{k} D\nabla S_w \cdot \nabla \psi \right) + \sum_{e \in \Gamma_D} \int_e \left( (\lambda_w \frac{dp_w}{dS_w})^{k} D\nabla S_w \cdot \nabla \psi \right) [\psi] \\
= \sum_{K \in \mathcal{T}_h} \int_K \lambda_w^{k} D\nabla p_w^{k+1} \cdot \nabla \psi_K + \sum_{e \in \Gamma_D} \int_e \lambda_w^{k} D\nabla p_w^{k+1} \cdot n_e \psi + \sum_{e \in \Gamma_D} \int_e \sigma_{\pi}^{k+1} p_{\pi}^{k+1} \psi + \sum_{K \in \mathcal{T}_h} \int_K \left( (\lambda_w \frac{dp_w}{dS_w})^{k} D\nabla S_w \cdot \nabla \psi \right) + \sum_{e \in \Gamma_D} \int_e \lambda_w^{k} D\nabla p_w^{k+1} \cdot n_e \psi + \sum_{e \in \Gamma_D} \int_e \sigma_{\pi}^{k+1} p_{\pi}^{k+1} \psi [\psi],
\]

where the Dirichlet boundary condition is weakly enforced. Noting that the gradient of the capillary pressure \( \nabla p_c \) in (3.34) is replaced by (3.14), the term \( - \nabla \cdot (\lambda_w D\nabla p_w) \) is discretized by the NIPG method, and the term \( - \nabla \cdot (\lambda_e D\nabla p_e) \) is treated by the upwind scheme.
To recover the total velocity from the variational equation (3.35), the following reconstruction technique is used,

\[
u_t^{k+1} \cdot n_e = -\{\lambda_k D\nabla p^{k+1} \cdot n_e\} - \{(\lambda_n \frac{d p_e}{d S_w})^k D\nabla S_w^k \cdot n_e\}, \quad (3.36)
\]

\[
u_t^{k+1} |K = -\lambda_k D\nabla p^{k+1} |K - (\lambda_n \frac{d p_e}{d S_w})^k D\nabla S_w^k |K. \quad (3.37)
\]

Now, we present the computation procedure of the sequential algorithms of the MHFE-PDG method and the PDG-PDG method.

- The sequential algorithm of the MHFE-PDG method is illustrated by the following steps:

  1. Set \( k = 0, m = 0 \) and \( t = t_0 \), choose an initial time step \( \Delta t_0 \), prescribe an error tolerance \( \epsilon_{tol} \), and initialize \( S^0_w = S_w(t_0) \).
  2. Supposing that the saturation \( S^k_w \) at time \( t_k \) and the time step \( \Delta t_k \) are given. Solve the equations (3.16)-(3.18) to obtain the solutions, and give the total velocity \( u_t^{k+1} \) to the saturation equation (3.30) or (3.32).
  3. Solve the equation (3.30) or (3.32) with the Picard iteration strategy, as follows.
    (a) Supposing that the solution \( S^{k,m}_w \) at the \( m \)-th step of the Picard iteration is known, \( m \geq 0 \), (when \( m = 0 \), \( S^{k,0}_w = S^k_w \)).
    (b) Based on different boundary conditions, solve the equation (3.30) or (3.32), and obtain \( S^{k+1,m+1}_w \).
    (c) Check the error (3.33), if the criterion is satisfied, stop the iteration of the nonlinear approximation and go to (e), else go to (d).
    (d) Update the solution, \( S^{k+1,m+1}_w \), and set \( m = m + 1 \), then continue (a) to (c).
    (e) Set \( t_{k+1} = t_k + \Delta t_k \), and obtain the solution at time \( t_{k+1} \) as \( S^{k+1}_w = S^{k+1,m+1}_w \). If \( t_{k+1} = T \), then stop the loop and take \( (S^{k+1}_w, p^{k+1}, u_t^{k+1}) \) as the final solution, else set \( k = k + 1, t_k = t_{k+1} \) and \( m = 0 \), and choose new \( \Delta t_k \), then go to (4).
  4. Repeat the steps from (1) to (3) until the predetermined final time \( T \) of the simulation is reached.

- The computation of sequential PDG-PDG method proceeds in the following order:

  1. Set \( k = 0, m = 0 \) and \( t = t_0 \), choose an initial time step \( \Delta t_0 \), prescribe an error tolerance \( \epsilon_{tol} \), and initialize \( S^0_w = S_w(t_0) \).
  2. Supposing that the saturation \( S^k_w \) at time \( t_k \) and the time step \( \Delta t_k \) are given. Solve the equation (3.35), obtain the pressure \( p^{k+1}_w \).
  3. Reconstruct the total velocity \( u_t^{k+1} \) through (3.36) and (3.37), and give it to the saturation equation (3.30) or (3.32).
  4. Solve the equation (3.30) or (3.32) with the Picard iteration strategy, as follows.
    (a) Supposing that the solution \( S^{k,m}_w \) at the \( m \)-th step of the Picard iteration is known, \( m \geq 0 \), (when \( m = 0 \), \( S^{k,0}_w = S^k_w \)).
    (b) Based on different boundary conditions, solve the equation (3.30) or (3.32), and obtain \( S^{k+1,m+1}_w \).
    (c) Check the error (3.33), if the criterion is satisfied, stop the iteration of the nonlinear approximation and go to (e), else go to (d).
    (d) Update the solution, \( S^{k+1,m+1}_w \), and set \( m = m + 1 \), then continue (a) to (c).
(e) Set \( t_{k+1} = t_k + \Delta t_k \), and obtain the solution at time \( t_{k+1} \) as \( S_{w}^{k+1} = S_{w}^{k+1,m+1} \). If \( t_{k+1} = T \), then stop the loop and take \( (S_w^{k+1},p^{k+1},u_t^{k+1}) \) as the final solution, else set \( k = k + 1, t_k = t_{k+1} \) and \( m = 0 \), and choose new \( \Delta t_k \), then go to (5).

(5) Repeat the steps from (1) to (4) until the predetermined final time \( T \) of the simulation is reached.

If we ignore the error criterion described in step (e), iterate just one step of the Picard iteration on each time step and take the solution as \( S_{w}^{k+1} = S_{w}^{k+1,1} \), the algorithms discussed above will become totally explicit approximations. We will compare the Picard iteration and the explicit approximation in the third numerical example below.

3.4 Adaptive Strategy We first introduce an a posterior error estimator based on superconvergent patch recovery (SPR) technique [23] and then adaptive mesh generation algorithm based on constrained centroidal Voronoi-Delaunay tessellations (CCVDT) is illustrated. Although there is no definition of interface in the two-phase Darcy model for two-phase flow in porous media, high resolution is needed where the variation of saturation is large. According to this simple observation, we recover the gradient saturation \( G_h S_{w}^{k,m} \) from PDG approximations \( S_{w}^{k,m} \) using the SPR method. The computational aspects can be found in [23]. Then The recovery-type local error estimator \( \eta_T \) associated with the element \( K \in T_h \) is given by

\[
\eta_K = \| \nabla S_{w}^{k,m} - G_h S_{w}^{k,m} \|_{L^2(K)} .
\]

Thus the a posterior error estimate can be conducted as

\[
\eta^2 = \sum_{K \in T_h} \eta^2_K = \| \nabla S_{w}^{k,m} - G_h S_{w}^{k,m} \|_{L^2(\Omega)}^2 .
\] (3.38)

The fact that the gradient \( G_h S_{w}^{k,m} \) recovered by SPR enjoys the superconvergent property on CVDT meshes is analysed and illustrated by abundant numerical examples in [24]. As a consequence, the asymptotical exactness of the a posterior error estimate can be guaranteed.

Voronoi-Delaunay triangulation is a popular method for unstructured mesh generation. Let \( \{z_i\}_{i=1}^k \) be a finite set of points belonging to a domain \( \Omega \in \mathbb{R}^d \) and for each \( i \), the point set \( V_i \) is defined as: \( V_i = \{p \in \Omega : \|p - z_i\| \leq \|p - z_j\|, \forall j \neq i\} \), where \( \|\cdot\| \) are metrics associated with the spaces and the Euclidean \( L^2 \) norm is frequently used in the area of mesh generation. Thus \( V_i \) is called the Voronoi region corresponding to the point \( z_i \). The collection of all the Voronoi regions \( \{V_i\}_{i=1}^k \) forms a partition of \( \Omega \) and is known as the Voronoi tessellation of \( \Omega \) with respect to the generating points \( \{z_i\}_{i=1}^k \). The Delaunay triangulation of \( \{z_i\}_{i=1}^k \) is defined as the dual of the Voronoi tessellation [25–27]. Delaunay triangulation is optimal in many ways due to the fact that the circum-ball associated with each element does not contain any other point of the triangulation except for the degenerate cases.

The centroidal Voronoi tessellation (CVT) and its wide range of applications have been studied in [7–10]. Often, CVT provides optimal points placement with respect to a given density function. Its dual structure, the so-called centroidal Voronoi Delaunay triangulation (CVDT), results in a high quality Delaunay mesh.

Given a density function \( \rho(x) \) defined on a region \( V_i \), the mass centroid \( z^* \) of \( V_i \) is defined by

\[
z^* = \frac{\int_{V_i} x \rho(x) \, dx}{\int_{V_i} \rho(x) \, dx} .
\]

Thus, given \( k \) points \( z_i, i = 1, \ldots, k \), in the domain \( \Omega \), we can define their associated Voronoi regions \( V_i, i = 1, \ldots, k \), which forms a tessellation of \( \Omega \). On the other hand, given the regions \( V_i, i = 1, \ldots, k \), we can define their mass centroids \( z^*_i, i = 1, \ldots, k \).
**Definition 3.1** (CVT and CVDT [7]). Given the set of points \( \{ z_i \}_{i=1}^k \) in the domain and a positive density function \( \rho(x) \) defined on \( \Omega \), a Voronoi tessellation is called a centroidal Voronoi tessellation (CVT) if
\[
z_i = z_i^*, \quad i = 1, ..., k,
\]
and the generators of the Voronoi regions \( V_i, z_i \), are themselves the mass centroids of those regions. The dual Delaunay triangulation is referred to as the Centroidal Voronoi-Delaunay triangulation (CVDT).

For any tessellation \( \{ V_i \}_{i=1}^k \) of the domain \( \Omega \) and a set of points \( \{ z_i \}_{i=1}^k \) in \( \Omega \), we can define the following cost (or error or energy) functional:
\[
\mathcal{F}(\{ V_i \}_{i=1}^k, \{ z_i \}_{i=1}^k) = \sum_{i=1}^k \int_{V_i} \rho(x) \| x - z_i \|^2 \, dx. \tag{3.39}
\]

The standard CVTs along with their generators are critical points of this functional. Using the concept of cost functional, we have

**Definition 3.2** (CCVDT [7, 10]). Given the set of points \( \{ z_i \}_{i=1}^k \) in \( \Omega \), a density function \( \rho(x) \), and a constraint set \( P \), a Voronoi tessellation is called a constrained centroidal Voronoi tessellation (CCVT) if \( \{ \{ V_i \}_{i=1}^k, \{ z_i \}_{i=1}^k \} \) is a solution of the problem
\[
\min_{\{ z_i \}_{i=1}^k \in P, \{ V_i \}_{i=1}^k} \mathcal{F}(\{ V_i \}_{i=1}^k, \{ z_i \}_{i=1}^k).
\]
The dual Delaunay triangulation is referred to as the constrained centroidal Voronoi-Delaunay tessellation (CCVDT).

**Algorithm 3.3** (modified CCVDT [23]). Given a domain \( \Omega \), a density function \( \rho(x) \) defined on \( \Omega \) and an initial constrained Delaunay tetrahedral mesh,

1. predetermine a subset of generating points on the boundary via a lower dimensional CVT construction based on the pre-defined density function \( \rho(x) \);
2. construct the Voronoi regions for all interior points that are allowed to change their positions and compute the mass centers of the Voronoi regions based on the given density function \( \rho(x) \) and use the computed mass centers as generating points;
3. retriangulation the domain \( \Omega \) using constrained Delaunay tessellation with generating points; the resulting triangulation is the new \( T_h \);
4. if the triangulation \( T_h \) meets some convergence criterion, return \( T_h \) and terminate; otherwise, go to step 1.

Since the a posteriori error estimator based on superconvergent gradient recovery \( \eta_K \) approaches to the true error asymptotically, we first use \( \eta_K \) to modify the sizing field on vertices \( \{ z_i^\ell \}_{i=1}^{n_\ell} \) at the refinement level \( \ell \). And then with the help of equidistribution principle, density function \( \rho(\ell+1) \) will be defined based on the modified sizing field.

We define the permissable error \( \mathcal{E}_p \) by
\[
\mathcal{E}_p = \delta \sqrt{\frac{\| S_{k,m} \|^2}{N(\ell)} + \frac{\eta^2_0}{N(\ell)}},
\]
where \( N(\ell) \) is the number of elements in \( T_h(\ell) \), \( \| \cdot \| \) is the energy norm and \( \delta \) is a pre-assigned positive constant usually with a value less than 1. In order to minimize \( \eta^2_0 = \sum_{K \in T_h} \eta^2_K = \sum_{K \in T_h} \eta^2_K \)
\[ \| \nabla S_{w}^{k,m} - G_{h}S_{w}^{k,m} \|_{L^{2}(\Omega)}^{2} \text{, we need to distribute } \eta_{K}^{2} = \| \nabla S_{w}^{k,m} - G_{h}S_{w}^{k,m} \|_{L^{2}(K)}^{2} \text{ equally over all} \]
\[ \text{triangulations of } T_{h}(\ell). \text{ Thus we modify the size of a element by} \]
\[ \tilde{h}_{K}(\ell) = \frac{h_{K}(\ell)}{\eta_{K} / \mathcal{E}_{p}}. \]

We then uniquely determine a piecewise linear function \( h^{(\ell+1)} \) on \( \Omega \) such that for any vertex \( z_{i}^{(\ell)} \) of \( K^{(\ell)} \),
\[ h^{(\ell+1)}(z_{i}^{(\ell)}) = \sum_{K \in \mathcal{K}_{i}} \frac{\tilde{h}_{K}(\ell)}{N(K_{i})}, \]
where \( \mathcal{K}_{i} = \{ K \in T_{h}(\ell) \mid z_{i}^{(\ell)} \in K \} \) and \( N(K_{i}) \) is number of triangulations contained in the node patch \( K_{i} \). Using the energy equidistribution property of CVT [7], the density function for \( T_{h}(\ell+1) \) is finally defined by
\[ \rho^{(\ell+1)}(z_{i}^{(\ell)}) = \frac{1}{(h^{(\ell+1)}(z_{i}^{(\ell)}))^{4}}. \]

Algorithm 3.4 (CCVDT-based adaptive mixed-hybrid and penalty discontinuous Galerkin methods).
Given a tolerance \( TOL > 0 \), at each adaptation level \( \ell \)

- Generate an initial CVDT mesh \( T_{h} \) over domain \( \Omega \).
- Solve the model using mixed-hybrid and penalty discontinuous Galerkin methods on \( T_{h} \).
- Compute the error estimators \( \eta_{K} \) on \( T_{h} \).
- While \( \eta_{M} > TOL \) do
  - Using the error estimator \( \eta_{K} \) to update the density function \( \rho \).
  - Perform splitting/contraction edge and other operation involving edge swapping, smoothing and combined optimization to refine/coraseen the mesh according to the density function \( \rho \).
  - Using modified CCVDT to optimize the mesh and update \( T_{h} \).
  - Solve the model using mixed-hybrid and penalty discontinuous Galerkin methods on \( T_{h} \).
  - Compute the error estimators \( \eta_{K} \) on \( T_{h} \).
end while.

The adaptive algorithm will be examined in the fifth numerical example below.

§ 4 Numerical examples

In this section, we present five tests of the proposed method on two dimensional space. In the first test, we examine the experimental convergence order of our numerical scheme, and compare with the simple projection PDG-PDG method. The second test considers the so called quarter-five spot problem introduced in [3]. In the third test, we examine the Van Duijn-De Neef problem [28] in homogeneous and heterogeneous porous media, and compare the explicit scheme and the Picard iteration. In the fourth test, the barrier effect phenomenon in a discontinuous porous media is simulated. In the last test, we examine the adaptive algorithm for the quarter-five spot problem in the homogeneous medium. The parameters containing rock and fluid properties used in the simulations are summarized in Table 1. It is noted that no slope limiters are used in test problems 1 and 3, but in test problems 2, 4 and 5 the slope limiter described in [11] is used.
Test 1 for EOC

<table>
<thead>
<tr>
<th>porosity</th>
<th>$\phi = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>permeability</td>
<td>$D=[1, 0; 0, 1]$</td>
</tr>
<tr>
<td>viscosity</td>
<td>$\mu_n = 1, \mu_w = 1$</td>
</tr>
<tr>
<td>residual saturation</td>
<td>$S_{rw} = 0, S_{rn} = 0$</td>
</tr>
<tr>
<td>Brooks-Corey</td>
<td>$p_d = 1, \theta = 2$</td>
</tr>
</tbody>
</table>

Test 2 and Test 5

| porosity       | $\phi = 0.2$ |
| permeability   | $D = [1, 0; 0, 1] \times 10^{-11}$ |
| viscosity      | $\mu_n = 2.0 \times 10^{-3}, \mu_w = 5.0 \times 10^{-4}$ |
| residual saturation | $S_{rw} = 0.05, S_{rn} = 0.1$ |
| Brooks-Corey   | $p_d = 5 \times 10^{3} Pa, \theta = 2$ |

Test 3 case 3a

| porosity       | $\phi_I = 1\backslash 1\backslash 1, \phi_{II} = 1\backslash 1\backslash 1$ |
| permeability   | $D_I = k_I \times [1, 0; 0, 1], D_{II} = k_{II} \times [1, 0; 0, 1]$ |
| viscosity      | $\mu_n = 1\backslash 1\backslash 1, \mu_w = 1\backslash 1\backslash 1$ |
| residual saturation | $S_{rI, w} = S_{rI, n} = 0, S_{rII, w} = S_{rII, n} = 0$ |
| Brooks-Corey   | $p_d^I = 1\backslash 1\backslash 1, p_d^{II} = 1\backslash 1\backslash \sqrt{2}, \theta^I = 2\backslash 2\backslash 2, \theta^{II} = 2\backslash 2\backslash 2$ |

| porosity       | $\phi_I = 0.2, \phi_{II} = 0.2$ |
| permeability   | $D_I = [1, 0; 0, 1] \times 10^{-10}, D_{II} = [1, 0; 0, 1] \times 10^{-11}$ |
| viscosity      | $\mu_n = 1.0 \times 10^{-2}, \mu_w = 1.0 \times 10^{-3}$ |
| residual saturation | $S_{rI, w} = S_{rI, n} = 0, S_{rII, w} = S_{rII, n} = 0$ |
| Brooks-Corey   | $p_d^I = 1 \times 10^{4} Pa, p_d^{II} = 1.5 \times 10^{4} Pa, \theta^I = 2, \theta^{II} = 2$ |

Table 1: Parameters used in the numerical simulations

4.1 Test problem 1 In the first test problem, we examine the experimental convergence order (EOC) of our numerical scheme, and compare with the PDG-PDG method. The final time $T$ is set to 1, the mesh sizes and the time steps are given in Table 2. The explicit MHFE-PDG and PDG-PDG methods are used. The adaptive strategy is not used.

The exact solutions are given on square mesh $(0, 1)^2$:

$$p_w = \cos\left(\frac{\pi}{8}\right)(x + y) + \frac{\pi}{8}(t);$$

$$S_w = 1 - \sin\left(\frac{\pi}{8}\right)(x + y) + \frac{\pi}{8}(t).$$

In order to use the nonlinear coefficients of Brook-Coery model (2.4)-(2.7), the exact saturation (4.2) is designed to take value between 0 and 1, which implies that the residual saturations satisfy $S_{rw} = S_{rn} = 0$ and the effective saturation $S_w$ is equivalent to the phase saturation $S_{w}$. Both the pressure and saturation equations use the Dirichlet boundary conditions in this test. The other parameters in the model are given in Table 1, test 1 for EOC. The numerical results are listed in Table 2, in which the flux norm is defined as:

$$||u||_F^2 := \sum_{K \in T_h} \sum_{e \in \partial K} |K| \int_e |u \cdot n_e|^2.$$  

where $n_e$ is the outward unit normal vector to $\partial K$.

From the tables we can see that the MHFE-PDG method with three $H(div)$ spaces, $RT_1$ and $BDM_1$, can obtain the optimal-order convergence for the three variables. But the PDG-PDG method does not obtain the optimal order for the saturation and pressure. Although the velocity
Table 2: The $L^2$ error and convergence order for three variables at $T=1$. At each space refinement, we take $dt=1/N$, $N=60$, 300, 1500, 7500, respectively.

<table>
<thead>
<tr>
<th></th>
<th>MHFE($RT_1$)-PDG</th>
<th>MHFE($BDM_1$)-PDG</th>
<th>PDG-PDG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>$</td>
<td></td>
<td>p - p_h</td>
</tr>
<tr>
<td>$1/4$</td>
<td>9.762e-3</td>
<td>2.936e-3</td>
<td>3.065e-2</td>
</tr>
<tr>
<td>$1/8$</td>
<td>3.593e-3</td>
<td>1.44</td>
<td>7.036e-3</td>
</tr>
<tr>
<td>$1/16$</td>
<td>1.007e-3</td>
<td>1.83</td>
<td>1.556e-3</td>
</tr>
<tr>
<td>$1/32$</td>
<td>2.373e-4</td>
<td>2.09</td>
<td>4.138e-5</td>
</tr>
</tbody>
</table>

Table 3: The comparison of CPU times (in seconds) corresponding to the results in Table 2. Re. means velocity reconstruction.

<table>
<thead>
<tr>
<th></th>
<th>MHFE($RT_1$)-PDG</th>
<th>MHFE($BDM_1$)-PDG</th>
<th>PDG-PDG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>MHFE</td>
<td>PDG</td>
<td>MHFE</td>
</tr>
<tr>
<td>$1/4$</td>
<td>4</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>$1/8$</td>
<td>31</td>
<td>249</td>
<td>25</td>
</tr>
<tr>
<td>$1/16$</td>
<td>851</td>
<td>5492</td>
<td>623</td>
</tr>
<tr>
<td>$1/32$</td>
<td>14993</td>
<td>106975</td>
<td>11173</td>
</tr>
</tbody>
</table>

reconstructed by (3.36)-(3.37) has the first-order convergence under $L^2$ norm, its error under $|\cdot|_{\text{div}}$ is not convergent. This implies that the recovered velocity in PDG-PDG method does not keep the local mass conservation property. Since the MHFE-PDG method and PDG-PDG method use the same PDG schemes for the saturation equation except for the different total velocities, the low order convergence of the saturation in PDG-PDG method must be caused by the simply reconstructed velocity which is low accurate and not locally balanced, then the convergence order of the pressure variable is also impaired.

The CPU times and the degrees of freedom of each method corresponding to the numerical experiments in Table 2 are recorded in Table 3 and Table 4 respectively. The recorded CPU times include the time cost by assembling the stiffness matrix and load vector and solving the corresponding algebraic system on all the time steps. For the second PDG method in PDG-PDG the time consumed by the velocity reconstruction is also recorded.

4.2 Test problem 2. In test 2, the so-called quarter-five spot problem [3] in a homogeneous medium is examined. The quarter-five spot problem simulates a process of water driving oil in the petroleum reservoir, that is, the water is injected from the injection well and drives the oil out of the production well. Based on this problem, we will compare the MHFE-PDG method and the PDG-PDG method, and check the error of local mass balance of each method.

The domain used in this experiment is a square $(0, 2)^2$ without two opposite corners; see Figure 2. The inflow boundary $\Gamma_{in}$ is located at the left bottom corner which represents an injection well, the outflow boundary $\Gamma_{out}$ is located at the right top corner which stands a production well, and
other boundary is the no-flow boundary $\Gamma_{no}$. The sink and source terms are set to zero inside the domain that is $q_w = 0$ and $q_n = 0$, because the production to the well is implemented by the boundary condition now. The following initial and boundary conditions are considered:

\begin{equation}
S_w(t = 0) = 0.2, \quad \text{on } \Omega
\end{equation}

\begin{equation}
S_n = 0.85 \quad \text{on } \Gamma_{in} \cup \Gamma_{in},
\end{equation}

\begin{equation}
\tilde{g}_N = 0 \text{ m/s}^{-1} \quad \text{on } \Gamma_{in} \cup \Gamma_{out},
\end{equation}

\begin{equation}
p_{dir} = 3.45 \times 10^6 \text{ Pa} \quad \text{on } \Gamma_{in},
\end{equation}

\begin{equation}
p_{dir} = 2.41 \times 10^6 \text{ Pa} \quad \text{on } \Gamma_{out},
\end{equation}

\begin{equation}
u_t \cdot n_e = 0 \quad \text{on } \Gamma_{no}.
\end{equation}

To make sure that the phase-front stays inside the domain, the final time is set to $T=80s$. The mesh size is $h = 0.25$. A constant time step is used, $dt = 80/1200$. The explicit algorithms are used for both combination methods, and the adaptive strategy is not used. For the $H(div)$ space we choose the $BDM_1$ space as an example, the space $RT_1$ have a similar performance.

The comparison of numerical results between the MHFE-PDG method and the PDG-PDG method at selected times are shown in Figures 2-3. From these Figures, we can see the drive process that the oil is gradually moving from the inflow boundary to the outflow boundary under the force of water drive. Although their differences in figures seems not so obvious, the error of local mass balance is very different between the MHFE-PDG method and the PDG-PDG method, see Table 5.

Since the sink and source terms are zero, the local mass is balanced on each element, that is, $\forall K \in T_h$, \[ \sum_{e \in \partial K} \int_e \mathbf{u}_t \cdot n_e = 0, \] where $n_e$ is the outward unit normal vector to $\partial K$. Thus we can easily define the errors of the local mass conservation in terms of the vector norms $l^\infty$ and $l^2$, which are defined respectively,

\begin{equation}
\max_{K \in T_h} \left( \left| \sum_{e \in \partial K} \int_e \mathbf{u}_t \cdot n_e \right| \right) \quad \text{and} \quad \left( \sum_{K \in T_h} \left| \sum_{e \in \partial K} \int_e \mathbf{u}_t \cdot n_e \right|^2 \right)^{\frac{1}{2}}.
\end{equation}

The errors of the local mass conservation of the two methods at selected times are listed in Table 5. We can see that the MHFE method almost exactly preserve the property of the local mass conservation up to the machine epsilon. However, the accuracy of the PDG method with reconstruction (3.36)-(3.37) is not very satisfactory. This result is consistent with the numerical results in the first test that the divergence error of the recovered velocity of the PDG-PDG method can not converge. So the reconstruction (3.36)-(3.37) is truly not enough accurate to give a locally balanced velocity.

4.3 Test problem 3 In this test, we consider the Van Duijn-De Neef problem which simulates a countercurrent flow in one dimensional heterogeneous media and studies the effect of the threshold capillary pressure on oil trapping [28]. Based on the PDG method for the saturation equation (4.19)-(4.22), the explicit iteration and the Picard iteration are compared. The adaptive strategy is not used.

<table>
<thead>
<tr>
<th>$h$</th>
<th>MHFE($RT_1$)</th>
<th>MHFE($BDM_1$)</th>
<th>PDG</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1/4$</td>
<td>80</td>
<td>80</td>
<td>96</td>
</tr>
<tr>
<td>$1/8$</td>
<td>352</td>
<td>352</td>
<td>384</td>
</tr>
<tr>
<td>$1/16$</td>
<td>1472</td>
<td>1472</td>
<td>1536</td>
</tr>
<tr>
<td>$1/32$</td>
<td>6016</td>
<td>6016</td>
<td>6144</td>
</tr>
</tbody>
</table>

Table 4: The comparison of degrees of freedom of each method corresponding to the results in Table 2.
The Van Duijn-De Neef problem provides a time-dependent semi-analytical solution, the so-called similarity solution which is a self-similar solution satisfying an ordinary differential equation (ODE) transformed from the original partial differential equation (PDE) of a two phase flow problem in one dimensional space [28]. In the countercurrent flow, the total velocity is equal to zero, so we consider the following one dimensional equations for the two phases in porous medium:

\[ \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda f_w k_l \frac{dp_c}{dS_w} \nabla S_w) = 0, \quad x < 0, \ t > 0, \quad (4.11) \]

\[ \phi \frac{\partial S_r}{\partial t} + \nabla \cdot (\lambda f_r k_r \frac{dp_c}{dS_w} \nabla S_w) = 0, \quad x > 0, \ t > 0, \quad (4.12) \]

where there is a single discontinuity at the origin, subscripts \( l \) and \( r \) stand the left side and right side of the origin, \( k_l \) (or \( k_r \)) is the absolute permeability on the left (or right) side. The flux of the solution should be continuous at the origin point,

\[ \lim_{x \to 0^+} \lambda_n f_w k_l \frac{dp_c}{dS_w} \frac{\partial S_w}{\partial x} = \lim_{x \to 0^-} \lambda_n f_r k_r \frac{dp_c}{dS_w} \frac{\partial S_w}{\partial x}. \quad (4.13) \]

The initial and boundary conditions for above equations are:

\[ S_w(x, 0) = \begin{cases} 
1, & x < 0, \\
0, & x > 0, 
\end{cases} \quad (4.14) \]

\[ S_w(-0.6, t) = 1, \quad S_w(0.6, t) = 0. \quad (4.15) \]

If we set \( S_w(x, t) = \psi(\eta) \) with \( \eta = \frac{x}{l} \), the original problem (4.11)-(4.15) can then be transformed into a boundary value problem consisting of ordinary differential equations, as follows,

\[ \eta \frac{d\psi}{d\eta} - k_l \frac{1}{\lambda_n f_w} \frac{d\psi}{d\eta} = 0, \quad \eta < 0 \quad (4.16) \]

\[ \eta \frac{d\psi}{d\eta} - k_r \frac{1}{\lambda_n f_r} \frac{d\psi}{d\eta} = 0, \quad \eta > 0 \quad (4.17) \]

\[ \psi(-\infty) = 1, \quad \psi(\infty) = 0. \quad (4.18) \]

The ODEs (4.16)-(4.18) are equivalent to the original PDEs (4.11)-(4.15), and with proper interface conditions of flux, capillary pressure and saturation this ODE system leads to a unique solution called similarity solution. And we will use this semi-analytical solution to examine the accuracy of our numerical scheme in this problem. For more details of this semi-analytical solution, see the reference [28].

In this experiment, we consider the two-dimensional extension of Van Duijn-De Neef problem on a narrow domain see Figure 4, and compare the numerical solutions derived by our scheme with the semi-analytical similarity solution in one dimension. The domain \( \Omega = [-0.6, 0.6] \times [0, h] \) is divided into two disjoint sub-domains, \( \Omega = \Omega_l \cup \Omega_{II} \), where the sub-domain on the left side is \( \Omega_l = [-0.6, 0] \times [0, h], \) and the right part is \( \Omega_{II} = [0, 0.6] \times [0, h] \). The interface between \( \Omega_l \) and \( \Omega_{II} \) is \( \Gamma_j = [0, 0] \times [0, h], \) where \( h = 1.2/n \), \( n \) is the even number of segments equally divided on the horizontal direction. The extension problem of (4.11)-(4.15) reads as:

\[ \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (\lambda f_w D \frac{dp_c}{dS_w} \nabla S_w) = 0, \quad (x, y, t) \in \Omega \times [0, T], \quad (4.19) \]
\begin{align}
S_w(-0.6, y, t) &= 1, S_w(0.6, y, t) = 0, \quad y \in [0, h], t \in [0, T], \\
\nabla S_w(x, 0, t) \cdot n &= \nabla S_w(x, h, t) \cdot n = 0, \quad x \in [-0.6, 0.6], t \in [0, T], \\
S_w(x, y, 0) &= 1, (x, y) \in \Omega_I, \quad S_w(x, y, 0) = 0, (x, y) \in \Omega_{II},
\end{align}

where the time is set to T=3 in the tests.

Since the total velocity is equal to zero ($u_i = 0$) in this problem, thus the advective term in (4.19) vanishes. The parameter values for the porous medium and Brooks-Corey model used in this test are listed in Table 1, in which the threshold capillary pressure $p_d$ is assumed to be $\sqrt{\phi/k}$, and $k$ is the scalar value of the absolute permeability.

**Test case 3a.**

The test case 3a is a homogeneous problem with the same rock property on each side of the interface, and the capillary pressure and the wetting phase saturation are always continuous on the interface $\Gamma_J$.

**Test case 3b.**

In the test case 3b, the left sub-domain $\Omega_I$ is the finer side with a greater absolute permeability than the right sub-domain $\Omega_{II}$. In this case, the capillary pressure is always continuous, and the wetting phase saturation is discontinuous. Furthermore, the critical point in (2.19) is $S_w^* = 0.5$, and $S_w^{II} \approx 0.33 < S_w^*, S_w^{II} < S_w^*$.

**Test case 3c.**

The third case is opposite to the previous one that the left sub-domain $\Omega_I$ becomes the coarse side, and $\Omega_{II}$ is the fine side. In this case, the critical point in (2.19) is $S_w^* = 0.64$, and $S_w^{II} \approx 0.4184 < S_w^*$, thus the capillary pressure is continuous. The wetting phase saturation is discontinuous and $S_w^{II} < S_w^*$ at the interface.

**Test case 3d.**

In the last case, based on the previous case, we enhance the jump of the permeability at the interface, so that the capillary pressure becomes discontinuous, and the critical point becomes $S_w^* = 0.25$ and $S_w^{II} \approx 0.4589 > S_w^*$. The wetting phase saturation is still discontinuous and $S_w^{II} = 1$ at the interface.

In all cases 3a-3d, the diffusion flux $\lambda_n f_w D \frac{\partial w}{\partial x} \nabla S_w \cdot n_e$ is always continuous across the interface $\Gamma_J$. The interface condition of wetting-phase saturation on $\Gamma_J$ is subject to what described in Subsection 2.2 when the permeability is discontinuous in cases 3b-3d. We compare the solution solved by implicit Picard iteration and by explicit approximation with the semi-analytical solution at three different times $t_1 = 1, t_2 = 2, t_3 = 3$. The mesh size is fixed to $h = 1.2/n$, $n = 64$, and three time steps $dt = \sqrt{h}$, $dt = h$, $dt = h^2$ are used. The tolerance error is set to $\epsilon_{tot} = 10^{-6}$. The results for the cases 3a-3d are presented in Figure 6, in which all the two-dimensional solutions are restricted to the line $y=0$. It is noted that the restriction to any $y \in [0, h]$ of the numerical two-dimensional solution is almost the same. The information of the iteration number and the computational time for the Picard iteration and the explicit iteration corresponding to the results in Figure 5 is given in Table 6.

For the explicit algorithm, the convergence is bad in all the four cases when the time step is $dt = \sqrt{h}$. As the time step gets smaller, the approximation is going to get better and better. Especially, when the time step is $dt = h^2$, the explicit scheme is almost as accurate as the Picard iteration. For the Picard iteration, in all cases the approximate solution converges to the semi-analytical solution. The iteration number of the Picard iteration decreases with reducing the time step size. At the same time step size, the Picard iteration is of course less efficient than the explicit
algorithm, since it has more iteration steps on each time step. So, in order to balance efficiency with accuracy, the explicit scheme should use a smaller time step, and the Picard iteration should use a bigger time step.

<table>
<thead>
<tr>
<th>case a</th>
<th>case b</th>
</tr>
</thead>
<tbody>
<tr>
<td>iteration number</td>
<td>CPU times(s)</td>
</tr>
<tr>
<td>ave.</td>
<td>max.</td>
</tr>
<tr>
<td>$dt = \sqrt{h}$</td>
<td>1</td>
</tr>
<tr>
<td>$dt = h$</td>
<td>1</td>
</tr>
<tr>
<td>$dt = h^2$</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>case c</th>
<th>case d</th>
</tr>
</thead>
<tbody>
<tr>
<td>iteration number</td>
<td>CPU times(s)</td>
</tr>
<tr>
<td>ave.</td>
<td>max.</td>
</tr>
<tr>
<td>$dt = \sqrt{h}$</td>
<td>1</td>
</tr>
<tr>
<td>$dt = h$</td>
<td>1</td>
</tr>
<tr>
<td>$dt = h^2$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 6: The comparison of the Picard iteration and the explicit iteration in four cases with different time steps. These data are corresponding to the numerical solutions at time $t_3$ in Figure 5. Pic., Exp., ave., and max. stand the Picard iteration, the explicit iteration, the average iteration number, and the maximum iteration number, respectively.

![Discontinuous quarter-five spot problem](image)

Figure 6: Discontinuous quarter-five spot problem

### 4.4 Test problem 4

In the fourth test, we simulate the quarter-five spot problem in discontinuous media. It includes the process of the oil-trapped phenomenon when the oil is flowing through the interface from the coarse medium to the fine medium. The implicit MHFE($RT_1$)-PDG method is used to simulate this problem, and no adaptive strategy is used.

We use a problem similar with that described in [13], which is designed to simulate a barrier effect of a nonwetting phase across a material interface. The domain used in the test is the square $(0, 2)^2$ with two small corners be cut off, which is initially fully water saturated and with the interfaces between two different sands, see Figure 6. When the oil flows from coarse sand to fine sand with the injection of oil from the inflow boundary $\Gamma_{in}$, more and more oil approaches and accumulates at the front of the interface of the fine sand. When the accumulation reaches a critical point, that is, when the capillary pressure at the coarse side of the interface is greater than the threshold pressure at the fine side, the accumulated oil will penetrate the interface and enter the fine sand area. By contrast, in the reversed situation the oil immediately penetrates the interface, that is, the oil-trapped phenomenon will not happen if the oil flows from fine sand to coarse sand.

In the following two cases the barrier effect phenomenon will be examined. The parameter values
for the porous media and Brooks-Corey model used in the tests are listed in Table 1. The Neumann-
Dirichlet type boundary (2.12)-(2.13) is used for the saturation equation, and the following initial
and boundary conditions are considered in the two cases:

\[
S_w(t = 0) = 1, \quad u_w \cdot n_e = 0 \text{ ms}^{-1} \quad \text{on} \quad \Gamma_{sN}(\Gamma_{in} \cup \Gamma_{no}), \quad (4.23) \\
S_{dir} = 1 \quad \text{on} \quad \Gamma_{sD}(\Gamma_{out}), \quad (4.24) \\
u_t \cdot n_e = 2.05 \times 10^{-2} \text{ ms}^{-1} \quad \text{on} \quad \Gamma_{pN}(\Gamma_{in}), \quad (4.25) \\
u_t \cdot n_e = 0 \text{ ms}^{-1} \quad \text{on} \quad \Gamma_{pN}(\Gamma_{no}), \quad (4.26) \\
p_{dir} = 2.01 \times 10^5 \text{ Pa} \quad \text{on} \quad \Gamma_{PD}(\Gamma_{out}). \quad (4.27)
\]

**Test case 4a**

In this case, \( \Omega_{II} \) is the fine sand and \( \Omega_I \) is the coarse sand, so the oil-trapped phenomenon will appear on the interfaces \( \Gamma_{II}^+ \), see Figure 6. The critical point in (2.19) is \( S_w^* \approx 0.44 \), and the oil will penetrate the interface \( \Gamma_{II}^+ \) when \( S_{II}^w \leq S_w^* \).

**Test case 4b**

In this case, we reverse the rock properties on the two sub-domains, that is \( \Omega_I \) is the fine sand and \( \Omega_{II} \) is the coarse sand. Thus the oil-trapped phenomenon will appear on the interfaces \( \Gamma_{II}^- \). The oil will penetrate the interface \( \Gamma_{II}^- \) when \( S_{II}^w \leq S_w^* \), where the critical point is also \( S_w^* \approx 0.44 \).

To make sure that the phase-front stays inside the domain, the final time is set to \( T=180s \) for both
cases. The mesh size is \( h = 0.125 \), and a constant time step \( dt = 0.1 \) is used. The tolerance error of
the Picard iteration for the saturation equation (3.32) is set to \( \epsilon_{tol} = 10^{-4} \). The numerical results at
selected times are drawn in Figure 7. In test case 4a, the average, minimum and maximum iteration
numbers are 4.077, 2 and 24 respectively. In test case 4b, the average, minimum and maximum
iteration numbers are 5.329, 2 and 32 respectively.

In test case 4a, \( \Omega_{II} \) is less permeable and has a higher entry pressure, so when the oil reaches the
interface at \( t \approx 40s \), it preferentially gets around \( \Omega_I \) since to enter in this sub-domain the pressure
must surpass the entry pressure \( p_{JII}^* = 1.5 \times 10^4 \text{ Pa} \). When this occurs at \( t \approx 60s \), only small fraction
of the oil enters \( \Omega_{II} \). At \( t = 180s \) the phase front still stays inside \( \Omega_{II} \) and does not reach the
production well. In test case 4b, in contrast to the previous test case, the oil directly enters \( \Omega_{II} \)
which has higher permeability. At approximately \( t \approx 180s \) the phase front completely turns out \( \Omega_{II} \). From
the comparison of the two cases, we see that contrast discontinuous permeability and capillary
pressure barriers can significantly influence on the secondary oil recovery process, in particular on
the flowing path and arriving time.

**4.5 Test problem 5**

In the last test, we examine our CCVDT-based adaptive mixed-hybrid
and penalty discontinuous Galerkin algorithm with the quarter-five spot problem in homogeneous
medium, and compare the results with the solutions on the uniformly refined mesh. The explicit
MHFE(\( RT_1 \))-PDG method is used to simulate this problem.

The domain is a square \((0,2)^2\) with two small corners be cut off. The spot at the left bottom
corner is the inflow boundary \( \Gamma_{in} \), the outflow boundary \( \Gamma_{out} \) is located at the right top corner,
and the other boundaries are the no-flow boundary \( \Gamma_{no} \). The same boundary and initial conditions
(4.4)-(4.9) are used. To make sure that the phase-front stays inside the domain the final time is set to
\( T=150s \). The mesh size of the uniformly refined mesh is about \( h \approx 0.0312 \). The constant time
step \( dt = 0.025 \) is used for the adaptive mesh and uniformly refined mesh respectively. The tolerance
is chosen to be \( TOL = 10^{-4} \). For this test problem, usually 2 or 3 adaptive iterations are needed at
each time step.
We compare the adaptive results with the results obtained on the uniformly refined mesh. The saturation contours at selected times are given in Figure 8. The contours on the adaptive meshes are similar to the uniformly refined ones. For better comparison, we also show the saturation profiles along the diagonal \{(x, y) : x = y\}. From the numerical results, we can see that the phase fronts can be captured by the adaptive meshes, where the elements near the phase front are smaller than uniform elements and the elements far away from the phase front are much coarser.

The solutions on the adaptive meshes are close to that on the uniform mesh near the phase fronts. Especially, when t=150s, the phase fronts are almost the same along the diagonal line, see Figure 8. However, the computational cost of adaptive algorithm is much lower than the uniform one, as shown in Table 7.

<table>
<thead>
<tr>
<th></th>
<th>Adaptive</th>
<th>Uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>t=60s</td>
<td>t=90s</td>
</tr>
<tr>
<td>$h_{\text{max}}$</td>
<td>0.1792</td>
<td>0.1782</td>
</tr>
<tr>
<td>$h_{\text{min}}$</td>
<td>0.0164</td>
<td>0.0153</td>
</tr>
<tr>
<td>Elements</td>
<td>2824</td>
<td>2681</td>
</tr>
<tr>
<td>CPU times(s)</td>
<td>119202</td>
<td>398087</td>
</tr>
</tbody>
</table>

Table 7: The comparison of the adaptive solution and the uniform solution. These data are corresponding to the numerical solutions in Figure 8.

§ 5 Conclusions

In this paper, we present a hybrid numerical method for the approximation of the fractional flow formulation of a two-phase flow problem, the combination of the mixed-hybrid and penalty discontinuous Galerkin method. This combination is rational and efficient, because, on one hand, the mixed-hybrid method is very suitable to the wetting phase pressure equation that is basically an elliptic equation with a continuous pressure variable and, more importantly, can provide an accurate total velocity as well as reduce the computational cost by utilizing less degrees of freedom and avoiding extra post-processing of velocity reconstruction, and on the other hand, the penalty discontinuous Galerkin method can cope with complicated interface conditions, which respects a discontinuous saturation. Some preliminary results of an adaptive algorithm for the hybrid method are presented. Our adaptive scheme is based on the centroidal Voronoi Delaunay triangulation (CVDT) technique and the resulting adaptive meshes are conforming. On the adaptive conforming meshes, the original limiters for the DG methods can still be used. Five numerical tests are presented to illustrate the proposed method. An optimal order of convergence can be achieved by our numerical method.

References


centroidal voronoi-delaunay tessellations for finite element methods. *Numerical Methods for


[26] D. F. Watson. Computing the n-dimensional delaunay tessellation with application to voronoi


[28] C. J. Van Duijn and M. J. De Neef. Similarity solution for capillary redistribution of two phases
in a porous medium with a single discontinuity. *Advances in water resources*, 21(6):451–461,
1998.
Figure 2: The wetting phase saturation of the MHFE($BDM_1$)-PDG method (left side) and the PDG-PDG method (right side) at selected times.
Figure 3: The wetting phase pressure of the MHFE(\textit{BDM}_1)-PDG method (left side) and the PDG-PDG method (right side) at selected times.
Figure 4: The mesh used for Test problem 3, the height $h = 1.2/n$, $n = 16$ here.

(a) case 3a $dt = \sqrt{h}$
(b) case 3a $dt = h$
(c) case 3a $dt = h^2$
(d) case 3b $dt = \sqrt{h}$
(e) case 3b $dt = h$
(f) case 3b $dt = h^2$
(g) case 3c $dt = \sqrt{h}$
(h) case 3c $dt = h$
(i) case 3c $dt = h^2$
(j) case 3d $dt = \sqrt{h}$
(k) case 3d $dt = h$
(l) case 3d $dt = h^2$

Figure 5: The comparison of the Picard iteration and the explicit iteration with the semi-analytical solution of the Van Duijn-De Neef problem at three different times ($t_1=1$, $t_2=2$ and $t_3=3$) in four cases with different permeabilities and capillary pressures and with three different time step sizes ($dt = \sqrt{h}$, $dt = h$, and $dt = h^2$). The mesh size is fixed to $h = 1.2/n$, $n = 64$. 
Figure 7: The wetting phase saturation solved by the MHFE($RT_1$)-PDG method at selected times for Test case 4a (left side) and Test case 4b (right side).
Figure 8: The wetting phase saturation solved by the MHFE($RT_1$)-PDG method on adaptive meshes (left) and uniformly refined mesh (middle), and the comparison along the diagonal line $x = y$ (right), at selected times.