

# A HIGH ORDER METHOD FOR THREE PHASE FLOW IN HOMOGENEOUS POROUS MEDIA

Justin Dong\*

Advisor: Béatrice Rivière\*

February 5, 2014

## Abstract

The modeling of three-phase fluid flow has many important applications in reservoir simulation. In this paper, we introduce a high order method for sequentially solving the phase pressure-saturation formulation of three-phase flow. The sequential approach, while less stable than a fully coupled approach, is more computationally efficient. We present a discontinuous Galerkin method in space, suitable for the sequential approach due to its high-order accuracy. We consider coarser meshes with high degree polynomials to minimize the dimension of the system while maximizing accuracy. Numerical results are given for homogeneous media.

## 1 Introduction

The modeling of three-phase flow – particularly of water, oil and gas – in porous media has many important applications, especially in the energy-related industry. Efficient extraction of fossil fuels is becoming increasingly crucial due to limited supply and increasing demand. The equations that model these phenomena are often extremely complex with no known analytical solution. Thus, the need for accurate and robust numerical solvers is a key component to effectively modeling three-phase flow in porous media.

In this paper, we discuss a high-order method for solving the pressure-saturation formulation of the three-phase flow equations. For the spatial discretization we use a discontinuous Galerkin (DG) method. The DG methods have recently gained popularity in multiphase flow simulations [9, 10, 14, 5] for three main reasons: (1) high degree of flexibility in regards to modifying the degree of the basis; (2) well-suitedness to handling the convective terms introduced in the three-phase formulation; and (3) local mass conservation, which is a crucial property for flow problems in porous media [13].

It is well known that the pressure-saturation formulation results in a strong coupling of pressure and saturations. Rather than solving the equations simultaneously, which can become very computationally expensive due to the necessity of computing a large Jacobian at each time step, we choose to decouple the equations and solve them

---

\*Department of Computational and Applied Mathematics, Rice University. This work was funded by Texas Norman Hackermann Advanced Research Program grant 003604-0015-2009

sequentially. More traditional methods for simulating multiphase flow include finite differencing schemes [12, 4, 3], classical finite element methods [6], and finite volume schemes [2]. These methods are of low order, typically no more than second order in accuracy. Furthermore, the sharp saturation fronts that are typical of numerical reservoir simulation [9] may not be well-resolved by these traditional methods, but are handled easily by the discontinuous basis functions employed by the DG method. The goal of our work is to investigate a more accurate and flexible method using coarser meshes and higher polynomial degrees. We consider varying the degree of the basis while keeping the global of the dimension of system constant. In this way, we can compare the computational efficiency of a high degree basis while keeping all other parameters constant.

This paper will focus on one-dimensional simulations. We first define the three-phase formulation in Section 2. The discrete formulation of the model problem is then given in Section 3 followed by the timestepping scheme and decoupling algorithm in Section 4. Finally, we present several numerical simulations in Section 5 followed by conclusions.

## 2 The Three-Phase Flow Formulation

The governing equations for incompressible three-phase flow in porous media are given by Darcy's law and the mass conservation equation [3, 12]:

$$\frac{\partial(\phi s_\alpha)}{\partial t} + \nabla \cdot \mathbf{u}_\alpha = q_\alpha \quad (2.1a)$$

$$\mathbf{u}_\alpha = -\frac{k k_{r\alpha}}{\mu_\alpha} \nabla p_\alpha \quad (2.1b)$$

where  $\alpha$  is the phase corresponding to water, oil and gas ( $w$ ,  $o$  and  $g$  respectively) and  $\phi$  and  $k$  are the porosity and absolute permeability of the medium. The variables  $\rho_\alpha$ ,  $\mathbf{u}_\alpha$  and  $\mu_\alpha$  are the density, volumetric flow rate, and viscosity of the phase. The saturation and pressure of the  $\alpha$  phase are given by  $s_\alpha$  and  $p_\alpha$ . Additionally,  $k_{r\alpha}$  is the relative permeability of the phase. Lastly,  $q_\alpha$  is the source term. It denotes an injection or leakage of the  $\alpha$ -phase from the system. We neglect gravity effects for the test cases in this paper.

From equations (2.1a) and (2.1b), we have three pairs of equations for three-phase flow. In order to obtain a numerical solution, we will use the phase pressure-saturation formulation in terms of a total velocity [8]. Before giving the phase pressure-saturation formulation, we define the phase mobility  $\lambda_\alpha$  and fractional flow  $f_\alpha$ . The total mobility  $\lambda$  is the sum of the phase mobilities. We have:

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha}, \quad \lambda = \lambda_w + \lambda_o + \lambda_g, \quad f_\alpha = \frac{\lambda_\alpha}{\lambda}$$

Choosing the primary variable for pressure to be the oil pressure, we introduce the water-oil and gas-oil capillary pressures  $p_{cwo}$  and  $p_{cgo}$  following [3]:

$$p_{cwo} = p_w - p_o, \quad p_{cgo} = p_g - p_o$$

The capillary pressures are functions of the phase saturations [8]. We express the total velocity  $\mathbf{u}$  – the sum of the phase velocities  $\mathbf{u}_\alpha$  – in terms of the oil pressure and oil capillary pressures:

$$\mathbf{u} = -k\lambda(\nabla p_o + f_w \nabla p_{cwo} + f_g \nabla p_{cgo}) \quad (2.2)$$

Using the definition that the sum of the saturations is unity and summing equation (2.1a) for all three phases, we obtain:

$$\nabla \cdot \mathbf{u} = \sum_{\alpha} q_{\alpha} \quad (2.3)$$

The initial condition and boundary conditions for the oil pressure are given by

$$\begin{aligned} p_o(t=0) &= p_o^0 \quad \text{in } \Omega \\ p_o &= p_{o,D} \quad \text{on } \partial\Omega \times [0, T] \end{aligned}$$

Here,  $\Omega$  denotes the computational domain and  $\partial\Omega$ , its boundary.  $T$  denotes the final time. The saturation equations are expressed in terms of  $\mathbf{u}$  and thus dependent on the oil pressure and capillary pressures. Furthermore, the presence of the  $\nabla(p_{c\beta o} - p_{c\alpha o})$  term increases the strength of the coupling of the system:

$$\frac{\partial(\phi s_{\alpha})}{\partial t} + \nabla \cdot \left( f_{\alpha} \mathbf{u} + k f_{\alpha} \sum_{\beta} \lambda_{\beta} \nabla(p_{c\beta o} - p_{c\alpha o}) \right) = q_{\alpha} \quad \text{in } \Omega \quad (2.4)$$

with boundary conditions and initial condition given by

$$\begin{aligned} s_{\alpha}(t=0) &= s_{\alpha}^0 \quad \text{in } \Omega \\ s_{\alpha} &= s_{\alpha,D} \quad \text{on } \partial\Omega \times [0, T] \end{aligned}$$

We consider the case of the pressure-saturation formulation where the capillary pressures are quadratic and increasing.

### 2.1 The Test Case: Quadratic Capillary Pressures

We take  $k = 1$  and  $\phi = 1$ . Additionally,  $q_w = q_o = q_g = 0$ . The phase mobilities are chosen as:

$$\lambda_w = s_w, \quad \lambda_g = s_g, \quad \lambda_o = 1 - s_w - s_g$$

We choose the following quadratic capillary pressures:

$$p_{cwo} = s_w^2 - 1, \quad p_{cgo} = s_g^2 - 1 \quad (2.5)$$

The three-phase flow equations corresponding to the quadratic capillary pressures take the following form:

$$-\frac{\partial^2 p_o}{\partial x^2} = \frac{\partial}{\partial x} \left( 2s_w^2 \frac{\partial s_w}{\partial x} \right) + \frac{\partial}{\partial x} \left( 2s_g^2 \frac{\partial s_g}{\partial x} \right) \quad (2.6a)$$

$$\frac{\partial s_w}{\partial t} + \frac{\partial}{\partial x} \left( \left( -\frac{\partial p_o}{\partial x} - 2s_w^2 \frac{\partial s_w}{\partial x} \right) s_w \right) - \frac{\partial}{\partial x} \left( 2s_w^2 (1 - s_w) \frac{\partial s_w}{\partial x} \right) = 0 \quad (2.6b)$$

$$\frac{\partial s_g}{\partial t} + \frac{\partial}{\partial x} \left( \left( -\frac{\partial p_o}{\partial x} - 2s_g^2 \frac{\partial s_g}{\partial x} \right) s_g \right) - \frac{\partial}{\partial x} \left( 2s_g^2 (1 - s_g) \frac{\partial s_g}{\partial x} \right) = 0 \quad (2.6c)$$

We can see that the first equation can be expressed as a diffusion equation for the oil pressure with a source term that is dependent on the water and gas saturations. The second and third equations are non-linear convection-diffusion equations. The presence of the pressure gradient in the convection coefficient causes a coupling of the saturations with the pressure.

Furthermore, the convection coefficient is directly related to the pressure gradient. For a large pressure gradient, the problem becomes convection-dominated. The conventional continuous finite element methods are ill-posed to handle the convective terms in this regard [11]. Thus, we choose the discontinuous Galerkin method in space, which is shown to possess superior stability properties for convection dominated problems [1].

### 3 The DG Formulation for a Convection-Diffusion Equation

Each of the saturation equations takes the form of a non-linear convection-diffusion equation:

$$\frac{\partial s(x, t)}{\partial t} + \frac{\partial}{\partial x} \left( \gamma \left( s, \frac{\partial s}{\partial x} \right) s(x, t) \right) - \frac{\partial}{\partial x} \left( \alpha(s) \frac{\partial s(x, t)}{\partial x} \right) = f(x, t) \quad x_0 \leq x \leq x_N \quad (3.1)$$

$$s(x_0, t) = g_0(t)$$

$$s(x_N, t) = g_N(t)$$

$$s(x, t_0) = s^0$$

Here, the convection coefficient  $\gamma$  is a function of both the solution  $s$  and its spatial derivative  $\frac{\partial s}{\partial x}$ . We define in the following sections a bilinear operator  $a(r; s, v)$  associated with the diffusion term and another bilinear operator  $b(r; s, v)$  associated with the convection term. We associate the right-hand side with the linear operator  $\ell(v)$ . Additionally, the  $L^2$  inner product on  $[x_0, x_N]$  is denoted by  $(\cdot, \cdot)$ . Thus, the discrete formulation is

$$\left( \frac{\partial s}{\partial t}, v \right) + b(s; u, v) + a(s; u, v) = \ell(v)$$

We partition the domain  $[x_0, x_N]$  into  $N$  intervals by introducing the interior nodes  $\{x_1, \dots, x_n, x_{n+1}, \dots, x_{N-1}\}$ . The subinterval  $[x_n, x_{n+1}]$  is denoted by  $I_n$ . We consider uniform meshes in this paper with mesh size  $h$ :

$$h = x_1 - x_0 = \dots = x_N - x_{N-1}$$

Let  $V_{DG}$  denote the space of piecewise discontinuous polynomials of degree  $p$ :

$$V_{DG} = \{v : v|_{I_n} \in \mathbb{P}^p(I_n), \forall n = 0, 1, \dots, N-1\}$$

We seek  $s_h \in V_{DG}$  such that for all  $v_h \in V_{DG}$ :

$$\left( \frac{\partial s_h}{\partial t}, v_h \right) + b(s_h; s_h, v_h) + a(s_h; s_h, v_h) = \ell(v_h) \quad (3.2)$$

#### 3.1 The Diffusion Term

We first define the jump and average as follows:

$$\{v(x_n)\} = \frac{1}{2}(v(x_n^+) + v(x_n^-))$$

$$[v(x_n)] = v(x_n^-) - v(x_n^+)$$

where  $v(x_n^+) = \lim_{x \rightarrow x_n^+} v(x)$  and  $v(x_n^-) = \lim_{x \rightarrow x_n^-} v(x)$ . The DG formulation of the diffusion term is given in [13]:

$$a(r; s, v) = \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} \alpha(r) s'(x) v'(x) dx - \sum_{n=0}^N \{\alpha(r) s'(x_n)\} [v(x_n)]$$

$$+ \epsilon \sum_{n=0}^N \{\alpha(r) v'(x_n)\} [s(x_n)] + \sum_{n=0}^N \frac{\sigma}{h} [s(x_n)] [v(x_n)] \quad (3.3)$$

In this bilinear form,  $\sigma$  is a penalty parameter introduced to “penalize” the jump

discontinuities of  $u$  and  $v$ . Additionally,  $\epsilon$  is a stabilizing parameter; for  $\epsilon = -1$ ,  $a(s; u, v)$  is symmetric and for  $\epsilon \in \{0, 1\}$ ,  $a(s; u, v)$  is non-symmetric. Note that for the simplicity of notation, we denote the spatial derivative of  $v$  by  $v'$ .

For the discretization of the three-phase formulation, we use the non-symmetric interior penalty Galerkin (NIPG) method with  $\epsilon = 1$  and  $\sigma = 1$  first introduced by Girault, Rivière and Wheeler in [7].

### 3.2 The Convection Term

The DG formulation of the convection term is obtained in the usual way by multiplying by a test function  $v \in V_{DG}$  and integrating over each interval. We introduce an upwinding scheme for the interior points of the mesh [1]:

$$\begin{aligned} \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} (\gamma(r, r')s(x))'v(x)dx &= - \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} \gamma(r, r')s(x)v'(x)dx \\ &\quad + \sum_{n=0}^N s^{up}(x_n)\{\gamma(r_n, r'_n)\}[v(x_n)] \\ s^{up}(x_n) &= \begin{cases} \lim_{\delta \rightarrow 0} s(x_n - \delta), & \{\gamma(r_n, r'_n)\} > 0 \\ \lim_{\delta \rightarrow 0} s(x_n + \delta), & \{\gamma(r_n, r'_n)\} < 0 \end{cases} \end{aligned}$$

The system is updated using information from the direction of the flow. For “upwind” (positive) convection, we update from the left. For “downwind” (negative) convection, we update from the right. Thus, the bilinear operator is:

$$b(r; s, v) = - \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} \gamma(r, r')s(x)v'(x)dx + \sum_{n=0}^N s^{up}(x_n)\{\gamma(r_n, r'_n)\}[v(x_n)] \quad (3.4)$$

### 3.3 The Right-Hand Side

Following [13], the right-hand side corresponding to the diffusion term is:

$$\begin{aligned} \ell(v) &= \int_{x_0}^{x_N} f(x, t)v(x)dx + \epsilon\alpha(s(x_0))v'(x_0)g_0(t) - \frac{\sigma}{h}v(x_0)g_0(t) \\ &\quad - \epsilon\alpha(s(x_N))v'(x_N)g_N(t) + \frac{\sigma^0}{h}v(x_N)g_N(t) \end{aligned} \quad (3.5)$$

Again, we add the stabilizing terms and the penalty terms in order to balance the corresponding terms in  $a(r; s, v)$ .

### 3.4 Discretization in Time

We implement a backward Euler method for the time discretization. Namely,

$$\left(\frac{s_h^k - s_h^{k-1}}{\Delta t}, v_h\right) + b(s_h^k; s_h^k, v_h) + a(s_h^k, s_h^k, v_h) = \ell(v_h) \quad (3.6)$$

## 4 A Decoupling Algorithm for Three-Phase Flow

We take the following steps to decouple the phase pressure-saturation equations. In equation (2.6a), let the source term in the right-hand side be denoted by  $f(s_w, s_g)$ . In order to solve for the oil pressure at the  $k^{\text{th}}$  time step,  $p_o^k$ , we use information from the  $k - 1$  time step:

$$-\frac{\partial^2 p_o^k}{\partial x^2} = f(s_w^{k-1}, s_g^{k-1}) \quad (4.1)$$

Thus, the oil pressure equation is now a linear elliptic problem. The gas and water saturation equations are each non-linear. Let  $b_g$  and  $b_w$  denote the bilinear forms of the convection term for the gas and water saturation equations, respectively. Similarly, let  $a_g$  and  $a_w$  denote the bilinear forms of the diffusion term for the gas and water saturation equations, respectively.

We first solve for  $p_o^k$  using  $s_w^{k-1}$  and  $s_g^{k-1}$ . Then, we use Newton's method to solve for  $s_w^k$  and  $s_g^k$ . The water saturation equation is:

$$\mathcal{F}_w(s_w^k) = \left( \frac{s_w^k - s_w^{k-1}}{\Delta t}, v_h \right) + b_w(s_w^k; s_w^k, v_h) - a_w(s_w^k; s_w^k, v_h) - \ell(v_h) \quad (4.2)$$

Since  $p_o^k$  is known, the water saturation equation is non-linear in  $s_w^k$  and  $\frac{\partial s_w^k}{\partial x}$ . Denote the  $i^{\text{th}}$  Newton iteration by  $(s_w^k)_i$ . Then  $(s_w^k)_{i+1}$  is written formally as:

$$(s_w^k)_{i+1} = (s_w^k)_i - \left( \frac{\partial \mathcal{F}_w}{\partial s_w^k} \right)^{-1} \mathcal{F}_w((s_w^k)_i)$$

The initial guess,  $(s_w^k)_0$ , is chosen to be the solution from the previous time step,  $s_w^{k-1}$ . Now let  $e_{s_w^k}^{i+1}$  denote the error in the  $(i + 1)^{\text{th}}$  Newton iterate at the  $k^{\text{th}}$  time step, i.e. the difference between the  $i^{\text{th}}$  and  $(i + 1)^{\text{th}}$  iterations:

$$e_{s_w^k}^{i+1} = |(s_w^k)_{i+1} - (s_w^k)_i|$$

The stopping criterion for the Newton loop is chosen to be when the error between successive iterates is less than a chosen tolerance value ( $10^{-10}$  in our numerical tests).

A similar case follows for the gas saturation equation:

$$\mathcal{F}_g(s_g^k) = \left( \frac{s_g^k - s_g^{k-1}}{\Delta t}, v_h \right) + b_g(s_g^k; s_g^k, v_h) + a_g(s_g^k; s_g^k, v_h) - \ell(v_h) \quad (4.3)$$

After solving for  $s_w^k$  and  $s_g^k$ , we compute  $f(s_w^k, s_g^k)$  and use this quantity to compute  $p_o^{k+1}$ . A template of the scheme is given in Algorithm 1.

Lastly, we need to consider the magnitude of the time step. One issue with using a backward Euler method in combination with a decoupling scheme is that the time step must be chosen wisely since using saturations from the previous time step to solve for the pressure is inherently less implicit. Choosing a time step that is too small is computationally inefficient while choosing a timestep that is too large may result in instability [4].

**Algorithm 1:** Decoupling Scheme

---

```

1 compute  $s_w^0$  and  $s_g^0$  using  $L^2$  projection;
2 compute  $f(s_w^0, s_g^0)$ , the source term for  $p_{o,DG}^1$  using  $s_{w,DG}^0$  and  $s_{g,DG}^0$ ;

3 /*  $k = T$  corresponds to the stopping time */
4 for  $k = 1, 2, \dots, T$  do
5     set tol and maxiter, the tolerance and maximum iterations for Newton's
      method;
6     initialize error iterates  $e_{s_w^k}$  and  $e_{s_g^k}$ ;
7     initialize iter1, iter2 = 0;
8     compute  $f(s_w^{k-1}, s_g^{k-1})$ ;
9     compute  $p_o^k$  using  $f(s_w^{k-1}, s_g^{k-1})$  and solving equation (4.1);

10    /* Newton loop */
11    while (iter1 < maxiter and  $e_{s_w^k} > \text{tol}$ ) do
12        compute  $s_w^k$  using  $p_o^k$  and  $s_w^{k-1}$  and solving equation (4.2);
13        compute error iterate  $e_{s_w^k}$  for the current Newton iteration;
14        iter1 = iter1 + 1;
15    end

16    /* Newton loop */
17    while (iter2 < maxiter and  $e_{s_g^k} > \text{tol}$ ) do
18        compute  $s_g^k$  using  $p_o^k$  and  $s_g^{k-1}$  and solving (4.3);
19        compute error iterate  $e_{s_g^k}$  for the current Newton iteration;
20        iter2 = iter2 + 1;
21    end
22 end

```

---

## 5 Numerical Results

We present several numerical simulations. First, we have implemented a discontinuous Galerkin method in space and backward Euler method in time for a single non-linear convection-diffusion equation modeled by equation (3.1). We compute errors and convergence rates for a known solution.

Next, we solve a system of three equations strongly resembling equations (2.6a)-(2.6c) with a known smooth solution. This is done by implementing a discontinuous Galerkin method in space, a backward Euler method in time and the decoupling scheme given in Algorithm 1. Errors and convergence rates are computed for known solutions.

Lastly, we simulate the three-phase flow equations from equations (2.6a)-(2.6c) for which there is no known analytical solution.

### 5.1 Example 1: A Non-linear Convection-Diffusion Problem

Consider the initial boundary value problem modeled by (3.1) on the domain  $[0, 1]$ . We take the following convection and diffusion coefficients:

$$\gamma\left(s, \frac{\partial s}{\partial x}\right) = -2s^2 \frac{\partial s}{\partial x}, \quad \alpha(s) = s + 2$$

We consider the case with the exact solution  $s(x, t) = t \sin x$ . We take the errors in the  $L^2$  and  $H^1$  norms defined as follows [13]:

$$\begin{aligned} \|s - s_h\|_{L^2(0,1)} &= \left( \int_0^1 (s - s_h)^2 dx \right)^{\frac{1}{2}} \\ \|s - s_h\|_{H^1(0,1)} &= \left( \sum_{n=0}^{N-1} \int_{x_n}^{x_{n+1}} (s' - s'_h)^2 dx \right)^{\frac{1}{2}} \end{aligned} \quad (5.1)$$

We perform a mesh refinement study with  $h \in \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$  and compute the convergence rates for bases of degrees  $p \in \{2, 3, 4\}$ .

<b>p</b>	<b>N</b>	$\ s - s_h\ _{L^2(0,1)}$	$L^2$ <b>Rate</b>	$\ s - s_h\ _{H^1(0,1)}$	$H^1$ <b>Rate</b>
2	2	$1.7540 \times 10^{-4}$	—	$1.8439 \times 10^{-3}$	—
	4	$2.8128 \times 10^{-5}$	2.6406	$4.5583 \times 10^{-4}$	2.0162
	8	$5.8992 \times 10^{-6}$	2.2534	$1.1270 \times 10^{-4}$	2.0160
	16	$1.4000 \times 10^{-6}$	2.0751	$2.7988 \times 10^{-5}$	2.0096
	32	$3.4437 \times 10^{-7}$	2.0233	$6.9725 \times 10^{-6}$	2.0051
3	2	$2.9321 \times 10^{-6}$	—	$1.5577 \times 10^{-5}$	—
	4	$1.5995 \times 10^{-7}$	4.1962	$1.8763 \times 10^{-6}$	3.0535
	8	$9.0028 \times 10^{-9}$	4.1511	$2.2115 \times 10^{-7}$	3.0848
	16	$5.1702 \times 10^{-10}$	4.1221	$2.6478 \times 10^{-8}$	3.0621
	32	$3.0558 \times 10^{-11}$	4.0806	$3.2267 \times 10^{-9}$	3.0367
4	2	$4.9240 \times 10^{-8}$	—	$4.7538 \times 10^{-7}$	—
	4	$1.6023 \times 10^{-9}$	4.9416	$3.0821 \times 10^{-8}$	3.9471
	8	$5.2191 \times 10^{-11}$	4.9402	$1.9443 \times 10^{-9}$	3.9866
	16	$1.8389 \times 10^{-12}$	4.8268	$1.2173 \times 10^{-10}$	3.9975
	32	$4.4163 \times 10^{-13}$	4.8646	$1.2325 \times 10^{-11}$	3.9665

Table 1: Convergence rates in the  $L^2$  and  $H^1$  norms for example 1. We take the time step  $\Delta t$  proportional to  $h^{p+1}$ , i.e.  $\Delta t = h^{p+1}$ .

For the NIPG method, we expect the convergence rates in the  $L^2$  and  $H^1$  norms to be  $p$  and  $p$ , respectively if  $p$  is even. If  $p$  is odd, then the convergence rates are  $p + 1$  and  $p$ , respectively [13]. As shown in Table 1, the convergence rates for the test problem conform to these expected rates for  $p = 2$  and  $p = 3$ . For  $p = 4$ , the convergence rate is actually super-optimal. Thus, the DG formulation for the non-linear convection-diffusion problem with the convection coefficient dependent on  $u$  and  $\frac{\partial u}{\partial x}$  exhibits convergence.

### 5.2 Example 2: A Non-linear, Coupled System

We now test the decoupling scheme on a system resembling the three-phase flow equations for linear capillary pressures. Our goal is to determine whether the decoupling

scheme still results in a convergent system, and if these convergence rates match what is expected by the NIPG method. Consider the system given by equations (2.6a)-(2.6c) on the domain  $[0, 1]$ . We modify the diffusion and convection coefficients for the purposes of this example. We modify the right-hand side of equations (2.6b) and (2.6c) to include non-zero source terms  $f_1(x, t)$  and  $f_2(x, t)$ :

$$-\frac{\partial^2 p_o}{\partial x^2} = \frac{\partial}{\partial x} \left( s_w \frac{\partial s_w}{\partial x} \right) + \frac{\partial}{\partial x} \left( s_g \frac{\partial s_g}{\partial x} \right) \quad (5.2a)$$

$$\frac{\partial s_w}{\partial t} + \frac{\partial}{\partial x} \left( \left( -\frac{\partial p_o}{\partial x} - s_w \frac{\partial s_w}{\partial x} \right) s_w \right) - \frac{\partial}{\partial x} \left( s_w (1 - s_w) \frac{\partial s_w}{\partial x} \right) = f_1(x, t) \quad (5.2b)$$

$$\frac{\partial s_g}{\partial t} + \frac{\partial}{\partial x} \left( \left( -\frac{\partial p_o}{\partial x} - s_g \frac{\partial s_g}{\partial x} \right) s_g \right) - \frac{\partial}{\partial x} \left( s_g (1 - s_g) \frac{\partial s_g}{\partial x} \right) = f_2(x, t) \quad (5.2c)$$

We choose the boundary conditions and initial condition to satisfy the exact solutions given by  $p_o(x, t) = \frac{1}{4} \cos(2xt) - \frac{1}{2} t^2 e^{2x}$ ,  $s_w(x, t) = te^x$  and  $s_g(x, t) = \sin(xt)$ . We obtain a numerical solution to the system by following the method in Algorithm 1.

The errors and convergence rates in the  $L^2$  and  $H^1$  norms are computed and given in Tables 2-4. We see that using the decoupled scheme still gives convergence. In fact, the convergence rates are actually super-optimal when  $p = 3$ : we expect the rates to be approximately 4.00 and 3.00 in the  $L^2$  and  $H^1$  norms, respectively, but the rates are actually about 4.00 and 4.00, respectively.

<b>p</b>	<b>N</b>	$\ p_o - p_{o,h}\ _{L^2(0,1)}$	$L^2$ Rate	$\ p_o - p_{o,h}\ _{H^1(0,1)}$	$H^1$ Rate
3	2	$2.4993 \times 10^{-3}$	—	$8.0765 \times 10^{-3}$	—
	4	$3.0377 \times 10^{-4}$	3.0405	$9.8261 \times 10^{-4}$	3.0390
	8	$1.9783 \times 10^{-5}$	3.9407	$6.4012 \times 10^{-5}$	3.9402
	16	$1.2907 \times 10^{-6}$	3.9380	$4.1784 \times 10^{-6}$	3.9373
4	2	$6.2500 \times 10^{-4}$	—	$2.0196 \times 10^{-3}$	—
	4	$3.8458 \times 10^{-5}$	4.0225	$1.2430 \times 10^{-4}$	4.0222
	8	$1.2201 \times 10^{-6}$	4.9782	$3.9439 \times 10^{-6}$	4.9781
	16	$3.8134 \times 10^{-8}$	4.9998	$1.2328 \times 10^{-7}$	4.9996

Table 2: Convergence rates in the  $L^2$  and  $H^1$  norms for  $p_o$  in example 2. We take the time step  $\Delta t$  proportional to  $h^{p+1}$ , i.e.  $\Delta t = h^{p+1}$ .

<b>p</b>	<b>N</b>	$\ s_w - s_{w,h}\ _{L^2(0,1)}$	$L^2$ Rate	$\ s_w - s_{w,h}\ _{H^1(0,1)}$	$H^1$ Rate
3	2	$1.1206 \times 10^{-4}$	—	$4.5640 \times 10^{-4}$	—
	4	$7.4396 \times 10^{-6}$	3.9129	$3.6657 \times 10^{-5}$	3.6382
	8	$4.6405 \times 10^{-7}$	4.0029	$2.3991 \times 10^{-6}$	3.9335
	16	$3.0048 \times 10^{-8}$	3.9489	$1.7381 \times 10^{-7}$	3.7869
4	2	$9.5746 \times 10^{-6}$	—	$4.3683 \times 10^{-5}$	—
	4	$2.8184 \times 10^{-7}$	5.0863	$1.6502 \times 10^{-6}$	4.7264
	8	$8.6946 \times 10^{-9}$	5.0186	$5.1764 \times 10^{-8}$	4.9945
	16	$2.7074 \times 10^{-10}$	5.0051	$1.6348 \times 10^{-9}$	4.9848

Table 3: Convergence rates in the  $L^2$  and  $H^1$  norms for  $s_w$  in example 2. We take the time step  $\Delta t$  proportional to  $h^{p+1}$ , i.e.  $\Delta t = h^{p+1}$ .

$\mathbf{p}$	$\mathbf{N}$	$\ s_g - s_{g,h}\ _{L^2(0,1)}$	$L^2$ Rate	$\ s_g - s_{g,h}\ _{H^1(0,1)}$	$H^1$ Rate
3	2	$2.9475 \times 10^{-5}$	—	$1.3376 \times 10^{-4}$	—
	4	$2.0186 \times 10^{-6}$	3.8681	$1.1479 \times 10^{-5}$	3.5426
	8	$1.2527 \times 10^{-7}$	4.0102	$7.2632 \times 10^{-7}$	3.9822
	16	$8.1848 \times 10^{-9}$	3.9306	$4.7633 \times 10^{-8}$	3.9306
4	2	$1.5888 \times 10^{-6}$	—	$8.5694 \times 10^{-6}$	—
	4	$4.8969 \times 10^{-8}$	5.0200	$3.4245 \times 10^{-7}$	4.6452
	8	$1.5012 \times 10^{-9}$	5.0276	$1.0634 \times 10^{-8}$	5.0092
	16	$4.6902 \times 10^{-11}$	5.0004	$3.3199 \times 10^{-10}$	5.0014

Table 4: Convergence rates in the  $L^2$  and  $H^1$  norms for  $s_g$  in example 2. We take the time step  $\Delta t$  proportional to  $h^{p+1}$ , i.e.  $\Delta t = h^{p+1}$ .

### 5.3 Example 3: Three-Phase Flow Simulation for Quadratic Capillary Pressures

We now solve the system given by equations (2.6a)-(2.6c). The initial conditions are visualized in Figure 1 given by the following distributions:

$$p_o(x, 0) = 200 - 199x \quad (5.3)$$

$$s_w(x, 0) = \begin{cases} 0.8 + \frac{0.5}{0.1}x, & x < 0.1 \\ 0.3, & x > 0.1 \end{cases} \quad (5.4)$$

$$s_g(x, 0) = \begin{cases} 0.1 + \frac{0.4}{0.1}x, & x < 0.1 \\ 0.5, & x > 0.1 \end{cases} \quad (5.5)$$

The boundary conditions of the pressure and saturations are

$$\begin{aligned} p_o(0, t) &= 200, & s_w(0, t) &= 0.8, & s_g(0, t) &= 0.1 \\ p_o(1, t) &= 1, & s_w(1, t) &= 0.3, & s_g(1, t) &= 0.5 \end{aligned}$$

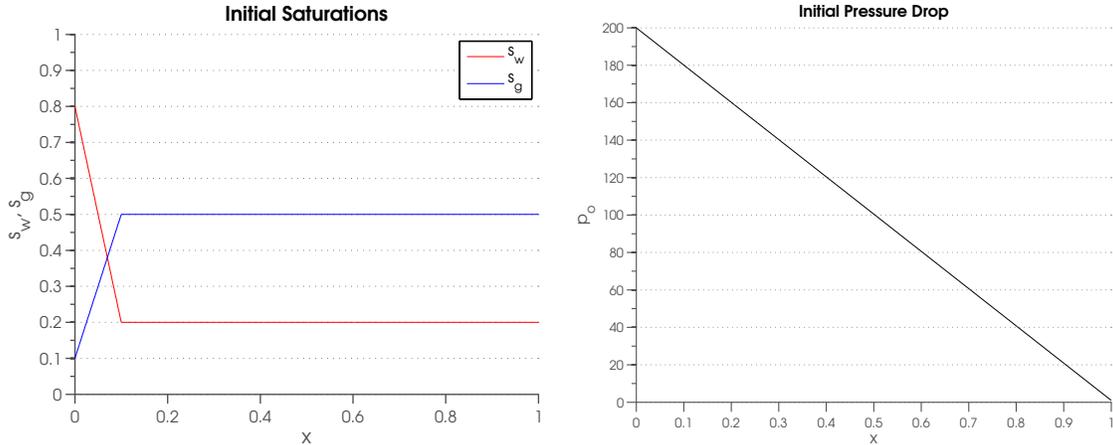


Figure 1: Initial conditions for the system of equations given by (2.6a)-(2.6c).

For this test, we take  $h = 1/5$  and  $p = 10$  with a time step of  $\Delta t = 0.00001$  s. Pressure and saturation curves are plotted at increments of every 30 time steps with a stopping time of 0.0036 s. The numerical results are given in Figures 2-4.

Next, we consider the global dimension of the system. For the discontinuous Galerkin method, the dimension is given by  $N(p + 1)$  where  $p$  is the degree of the polynomial basis and  $N$  is the number of mesh elements. Considering the case with  $p = 10$  and  $h = 1/5$ , the global dimension is 55. Fixing the time step at  $\Delta t = 0.00001$  s, we vary  $p$  and  $N$  while keeping the global dimension roughly constant. In this way, we might expect the computation times to be similar. However, as Figure 5 demonstrates, this is not the case.

As  $p$  increases and  $N(p + 1)$  stays constant, we see that the CPU time initially decreases drastically then begins to level off around  $p = 5$ . This seems to suggest that it is actually advantageous to use a higher degree basis with less mesh elements. However, for  $p$  larger than 10, poor conditioning of the local matrices increases the CPU time and causes instabilities in the solutions.

Next, we consider the accuracy of the solution on the coarse mesh,  $h = 1/5$  and  $p = 10$ . We take the solution on a much finer mesh,  $h = 1/40$  and  $p = 3$ , which has high accuracy, and compare it to the solution on the coarse mesh. In Figures 6-8, we superimpose the solutions at time  $t = 0.0012$ . As indicated by the results, the solution on the coarse mesh is just as accurate as the solution on a much finer mesh. Thus, our choice of  $h = 1/5$  and  $p = 10$  is not only computationally efficient but also stable and accurate.

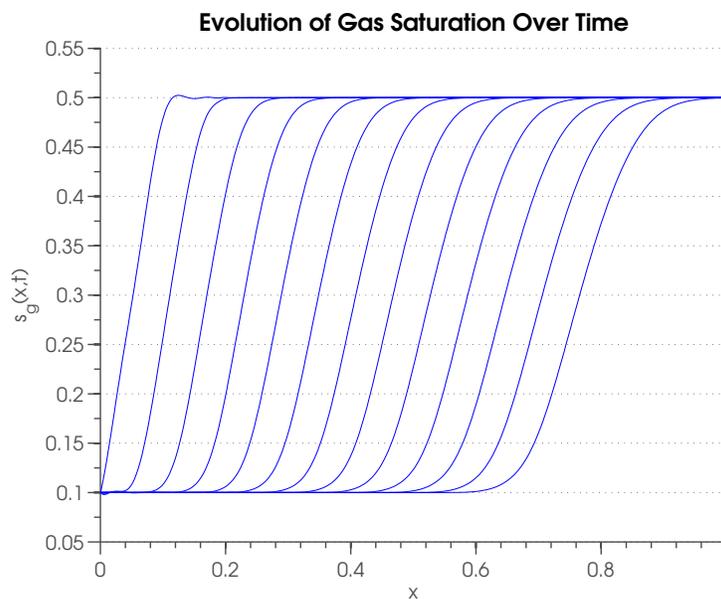


Figure 2: Numerical result for gas saturation with quadratic capillary pressures,  $\Delta t = 0.00001$ ,  $p = 10$  and  $h = 1/5$ . Plotted at every  $30^{th}$  time step.

## 6 Conclusions

In this paper, we have presented a discontinuous Galerkin discretization in space for simulating the phase pressure-saturation formulation of three-phase flow in one-dimension. Using a backward Euler method in time, we decouple the system and solve the equations sequentially. We show that the solution is convergent using this scheme as long as the time step is chosen carefully.

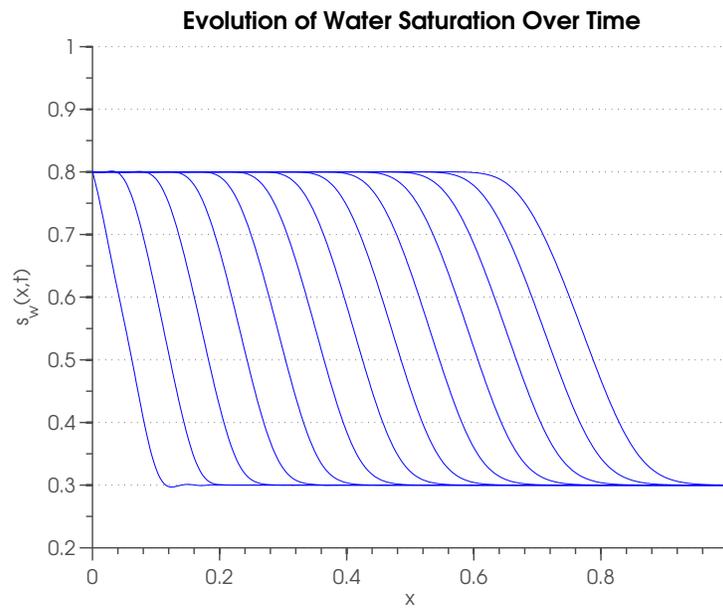


Figure 3: Numerical result for water saturation with quadratic capillary pressures,  $\Delta t = 0.00001$ ,  $p = 10$  and  $h = 1/5$ . Plotted at every  $30^{th}$  time step.

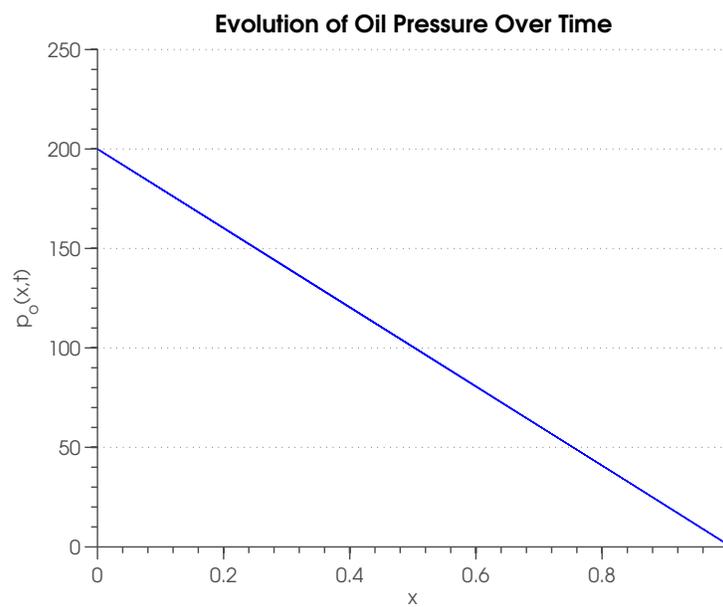


Figure 4: Numerical result for oil pressure with quadratic capillary pressures,  $\Delta t = 0.00001$ ,  $p = 10$  and  $h = 1/5$ . Plotted at every  $30^{th}$  time step.

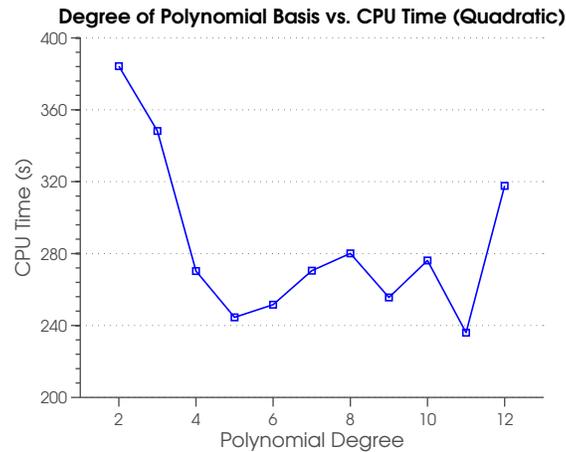
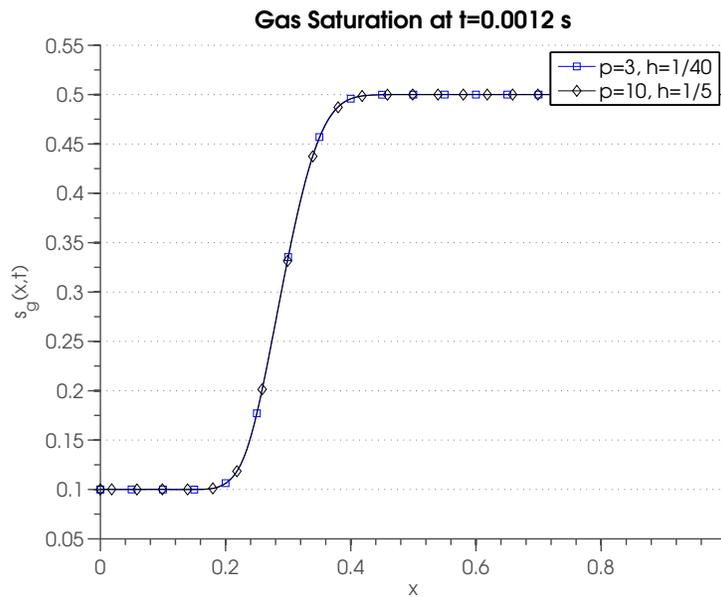


Figure 5: CPU times for quadratic capillary pressures.

Figure 6: A comparison of numerical results on the coarse mesh and fine mesh for gas saturation at  $t = 0.0012$  with quadratic capillary pressures.

Our simulations of the three-phase flow demonstrate that solutions on coarser meshes with higher polynomial degrees are actually more computationally efficient than solutions on finer meshes with the same global dimension. Furthermore, we compute the solution on a fine mesh and show that solution on the coarser mesh has a similar degree of accuracy.

Future work on this subject includes an adaptation to two-dimensions. Additionally, high-order discontinuous Galerkin methods are well-suited to GPU high-performance computing. Thus, we could develop an algorithm that is more computationally efficient in terms of CPU time.

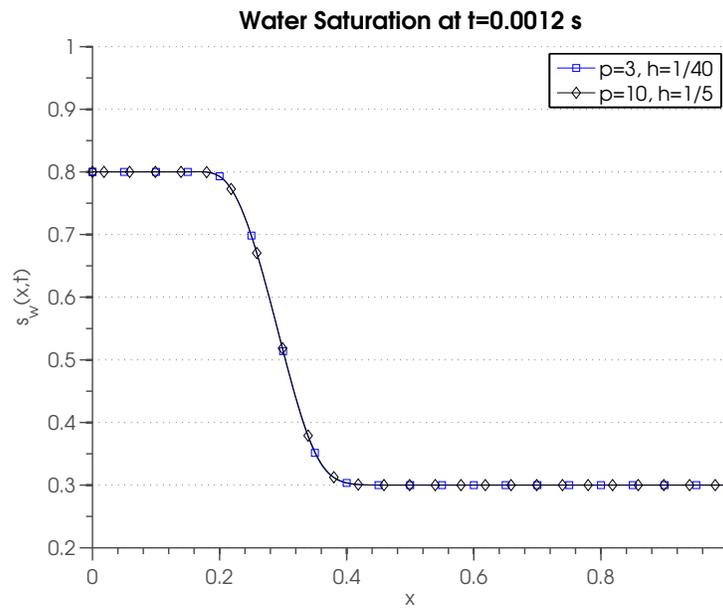


Figure 7: A comparison of numerical results on the coarse mesh and fine mesh for water saturation at  $t = 0.0012$  with quadratic capillary pressures.

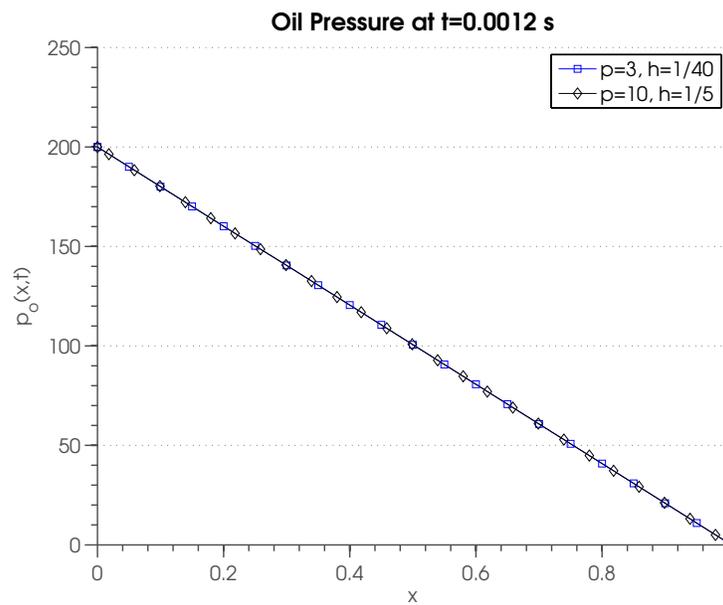


Figure 8: A comparison of numerical results on the coarse mesh and fine mesh for oil pressure at  $t = 0.0012$  with quadratic capillary pressures.

---

## References

- [1] C. E. BAUMANN AND J. T. ODEN, *A discontinuous hp finite element method for convection-diffusion problems*, Computer Methods in Applied Mechanics and Engineering, 175 (1999), pp. 311–341.
- [2] K. BRENNER, C. CANCES, D. HILHORST, ET AL., *Finite volume approximation for an immiscible two-phase flow in porous media with discontinuous capillary pressure*, Computational Geosciences, 17 (2011), pp. 573–597.
- [3] Z. CHEN AND R. EWING, *Comparison of various formulations of three-phase flow in porous media*, Journal of Computational Physics, 132 (1997), pp. 362–373.
- [4] Z. CHEN, G. HUAN, AND Y. MA, *Computational methods for multiphase flows in porous media*, vol. 2, Society for Industrial and Applied Mathematics, 2006.
- [5] Z. CHEN, H. STEEB, AND S. DIEBELS, *A space-time discontinuous Galerkin method applied to single-phase flow in porous media*, Computational Geosciences, 12 (2008), pp. 525–539.
- [6] R. E. EWING, *The mathematics of reservoir simulation*, Siam, 1983.
- [7] V. GIRAULT, M. RIVIÈRE, AND M. WHEELER, *Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptical problems*, Computational Geosciences, 3 (1999), pp. 337–360.
- [8] R. HELMIG, *Multiphase flow and transport processes in the subsurface: a contribution to the modeling of hydrosystems*.
- [9] W. KLIEBER AND B. RIVIÈRE, *Adaptive simulations of two-phase flow by discontinuous Galerkin methods*, Computer methods in applied mechanics and engineering, 196 (2006), pp. 404–419.
- [10] I. MOZOLEVSKI AND L. SCHUH, *Numerical simulation of two-phase immiscible incompressible flows in heterogeneous porous media with capillary barriers*, Journal of Computational and Applied Mathematics, 242 (2012), pp. 12–27.
- [11] J. T. ODEN AND C. E. BAUMANN, *A conservative DGM for convection-diffusion and Navier-Stokes problems*, in Discontinuous Galerkin methods, Springer, 2000, pp. 179–196.
- [12] D. PEACEMAN, *Fundamentals of numerical reservoir simulation*, vol. 6, Elsevier Science, 2000.
- [13] B. RIVIÈRE, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*, SIAM, 2008.
- [14] P. SOCHALA, A. ERN, AND S. PIPERNO, *Mass conservative BDF-discontinuous galerkin/explicit finite volume schemes for coupling subsurface and overland flows*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 2122–2136.