The Thesis Committee for Eric Alexander Lynd
Certifies that this is the approved version of the following thesis:

An Exploration of the IGA Method for Efficient Reservoir Simulation

APPROVED BY

SUPERVISING COMMITEE:

John Foster, Supervisor

Quoc Nguyen, Co-Supervisor

An Exploration of the IGA Method for Efficient Reservoir Simulation

by

Eric Alexander Lynd

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2017

Acknowledgments

I first want to thank Dr. John Foster and Dr. Quoc Nguyen for their continuous guidance and support over the course of my time at the University of Texas at Austin. Their dedication and expertise speak to the ongoing excellence of the Petroleum and Geosystems Engineering department, which I have been proud to be a part of these past years. For their camaraderie during my time at UT Austin, I also want to extend my gratitude to Brandon Tripp and Arin Raina. Their good humor and friendship made the long hours of study and work far more enjoyable. Lastly, for their unconditional love and encouragement, I give my sincerest thanks to my family. I hope to continue making them proud over the years ahead.

ERIC ALEXANDER LYND

The University of Texas at Austin
August 2017

An Exploration of the IGA Method for Efficient Reservoir Simulation

Eric Alexander Lynd, MSE
The University of Texas at Austin, 2017

Supervisors: John Foster, Quoc Nguyen

Novel numerical methods present exciting opportunities to improve the efficiency of reservoir simulators. Because potentially significant gains to computational speed and accuracy may be obtained, it is worthwhile explore alternative computational algorithms for both general and case-by-case application to the discretization of the equations of porous media flow, fluid-structure interaction, and/or production. In the present work, the fairly new concept of isogeometric analysis (IGA) is evaluated for its suitability to reservoir simulation via direct comparison with the industry standard finite difference (FD) method and 1st order standard finite element method (SFEM). To this end, two main studies are carried out to observe IGA's performance with regards to geometrical modeling and ability to capture steep saturation fronts. The first study explores IGA's ability to model complex reservoir geometries, observing L2 error convergence rates under a variety of refinement schemes. The numerical experimental setup includes an $'S'$ shaped line sink of varying curvature from which water is produced in a 2D homogenous domain. The accompanying study simplifies the domain to 1D, but adds in multiphase physics that traditionally introduce difficulties associated with modeling of a moving saturation front. Results overall demonstrate promise for the IGA method to be a particularly effective tool in handling geometrically difficult features while also

managing typically challenging numerical phenomena.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

It is the ultimate goal when designing reservoir simulation software to maximize the accuracy in depictions of physical phenomena while minimizing the computational expense. This overall efficiency is in large part controlled by the optimal selection of numerical methods, which may vary in terms of ease of applicability and rate of convergence on a case-by-case basis. In an ideal situation, a suite of numerical methods should be available for occasions when one alone presents intractable run times or unacceptable error at coarse scale discretizations. Below, geometrical and numerical motivations for exploration into alternative methods are discussed, along with a brief background of the recently developed isogeometric analysis method.

1.1 Motivation

1.1.1 Geometrical Modelling Limitations

Reservoir fluid transport in many formations is dominated by localized phenomena that require accurate geometric descriptions of the field equations for simulation codes to be predictive. As an example, consider either natural or hydraulically stimulated fractures (Berkowitz, 2002; McCord et al., 1992). The standard approaches for modeling fractures in commercial reservoir simulators are dual-porosity models (Fig. 1.1), which suffer shortcomings in the presence of fractures that are not well-connected, distributed non-uniformly, or which have a few large but connected fractures that dominate the flow (Gerke and Genuchten, 1993; Zimmerman et al., 1993). Additionally, they use *ad hoc* transfer functions to account for fluid exchange between the porous matrix and the fractures (Sahimi, 2011). Discrete fracture models (Fig. 1.2) have also been used which model the fractures explicitly, but require conforming finite element discretizations (Karimi-Fard et al., 2003; Matthäi et al., 2005). If piecewise constants (i.e., finite volumes) or the typical linear basis functions are used for the approximation of the solution over the elements, then a highly refined region near the fractures is required for accurate resolution of both steep pressure gradients and fracture geometry. These highly refined regions lead to impracticable computation times for models at reservoir

scales. Besides fractures, "snake wells" are another example where complex geometry manifests in reservoir engineering and traditional simulators are challenged to be predictive (Johan et al., 2004).

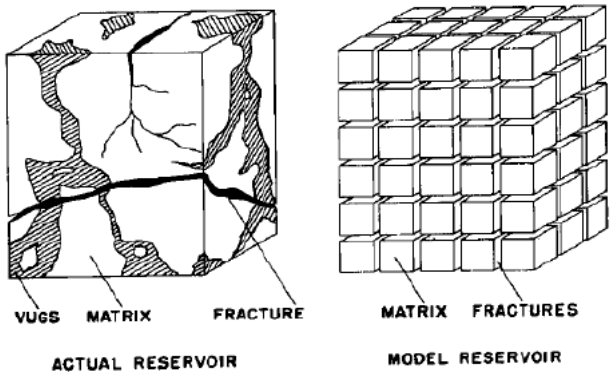


Figure 1.1: Dual-porosity diagram (Warren et al., 1963).

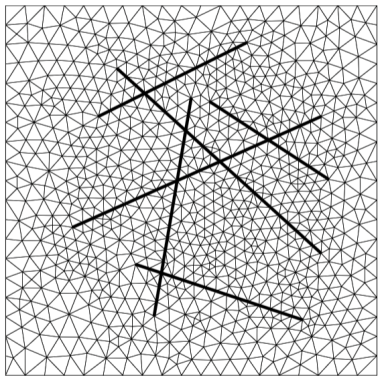


Figure 1.2: Example of discrete fracture network (Karimi-Fard et al., 2001).

1.1.2 Numerical Concerns Surrounding Steep Gradient Phenomena

Geometrical constraints in the implementation of numerical methods are often accompanied by numerical constraints. In particular, it has been found that when modeling steep gradient advection-diffusion type problems with the finite difference method, excess 'smearing' in the form of numerical diffusion occurs (Finlayson, 1992). As an example, a standard advection-dominated concentration profile modeled by FD is shown in Figure 1.3. Here, insufficient resolution of the FD grid leads to poor representation of the shock front via averaging of the field variable over non-physical distances. This of course obfuscates important flow information, chiefly the shock front concentration and shock front break through time. While advances have been made to address numerical diffusion associated with the finite difference method (Chen et al., 1991; Shu, 2003), global grid refinements and longer run times are still often used as a means to address local or global numerical issues.

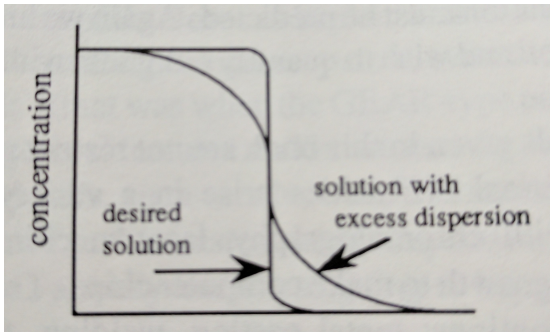


Figure 1.3: Example of numerical diffusion effecting the modeling of an advection-diffusion concentration front (Finlayson, 1992).

Higher order finite element methods are plagued by similar issues when confronted with steep gradient phenomena and poor grid resolution. However, these methods may also possess near-front, propagating oscillations (Pinder and Gray, 1977). Figure 1.4 shows an example of such undesired behavior. Severe deficiencies in the FE derived simulation for advection-dominated flows have contributed to developers relying heavily on finite difference and finite volume methods, despite FE's robust selection of basis functions and gridding schemes. Popular remedies to the issue of near shock oscillations include variational multiscale (VMS) methods (Brooks and Hughes, 1982; Hughes et al., 1998), adaptive gridding (Almeida et al., 2000), and extended finite element (XFEM) formulations (Abbas et al., 2010).

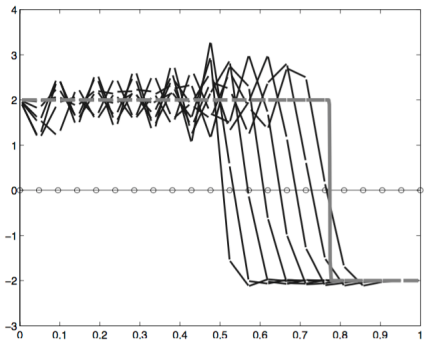


Figure 1.4: Example of oscillations effecting the modeling of an advection-diffusion concentration front (Abbas et al., 2010).

1.2 A Brief Overview of Isogeometric Analysis

There has been a rising popularity of non-uniform rational B-spline (NURBS) enhanced methods throughout multiple engineering fields (Sevilla et al., 2006; Tan et al., 2015). NURBS have the ability to *exactly* represent all conic sections and are the standard geometric interpolant used in computer aided design (CAD) software. Recently, isogeometric analysis (IGA) has been established as an alternative to finite-element anal-

ysis (FEA) and is novel for its use of NURBS basis functions in both geometry description and analysis (Hughes et al., 2005). Such treatment ameliorates the time cost typically associated with meshing, and allows for *exact* representation of complex geometries. In addition, IGA possesses a k-refinement strategy in which continuity of basis functions across elements can be increased while removing degrees of freedom. More recently developed features that have become available to IGA include local refinement with T-splines (Bazilevs et al., 2010), efficient use of Gauss quadrature points for high continuity basis functions (Adam et al., 2015), and a Bézier extraction operator that allows closer resemblance of an IGA code to finite element analysis (Borden et al., 2011). Particular application to the reservoir simulation field have so far revolved around fluid/fracture interaction in deformable media (Irzal et al., 2014; Shahrokhabadi et al., 2014; De Luycker et al., 2011).

1.3 Summary of Thesis

Broadly, the goal of this thesis is to further expose the petroleum engineering community to IGA and demonstrate its efficacy on simple reservoir problems where complex geometries or classically challenging numerical phenomena are present. In Chapter 2, relevant equations and formulations are outlined. Chapter 3 gives a more thorough treatment of IGA's fundamentals and details the evaluation and construction of NURBS functions. Next, general testing procedures and implemented parameters are outlined in Chapter 4. Chapter 5 and 6 contain the results and discussion of the several investigative studies, and are followed by concluding remarks in Chapter 7.

Chapter 2

Equations and Formulations

2.1 Fundamental Equations for Geometrical Study

Geometrical studies in the later sections of this paper involve the calculations of pressure fields across two-dimensional (2D), aerial view (i.e. no gravity), and isotropic porous media. Additionally, flow is idealized to be single-phase (water), Newtonian, steady-state, and incompressible. Line sinks of constant flux are used to simulate the effect of wells or fractures on the pressure field of a reservoir. The equations involved in the development of the numerical method studied in this work are summarized in the sequel.

2.1.1 Darcy's Law and Mass Balance

Darcy's Law for 2-D flow is given as

$$\vec{v}(x,y) = \frac{k}{\mu}\nabla P(x,y), \tag{2.1}$$

where k is the isotropic permeability, μ is viscosity, $\vec{v}$ is the velocity vector, ∇ is the spatial gradient operator, and P is the pressure field. Furthermore, the ratio of permeability to viscosity can be expressed as:

$$\lambda = \frac{k}{\mu}, \tag{2.2}$$

where λ is the mobility. Assuming incompressible, steady state flow, the mass balance equation is given as

$$\nabla \cdot \vec{v}(x,y) = 0. \tag{2.3}$$

Substituting (2.1) into (2.3) gives the partial differential equation governing the pressure distribution in the 2D domain Ω:

$$\nabla \cdot (\lambda \nabla P(x,y)) = 0. \tag{2.4a}$$

Finally, Dirichlet and Nuemann boundary conditions may be expressed as

$$\hat{P}(x,y) = P(x,y) \quad \text{for} \quad (x,y) \in \Gamma_D, \tag{2.4b}$$

$$\hat{v}(x,y) = \lambda \nabla P(x,y) \quad \text{for} \quad (x,y) \in \Gamma_N, \tag{2.4c}$$

where $\hat{P}$ and $\hat{v}$ are prescribed values for pressure and velocity, Γ_D is the Dirichlet boundary, and Γ_N is the Neumann boundary.

2.1.2 Discretization of Pressure Field Equation

A Ritz-Galerkin weak form for eqs. (2.4) is derived using an approach identical to that for the standard FEM case (Becker et al., 1981). First, (2.4a) is multiplied by a test function w and integrated:

$$\int_\Omega w \nabla \cdot (\lambda \nabla P) = 0. \tag{2.5}$$

Next, integration by parts is used to shift a derivative onto the test function and thereby "weaken" the requirements of solving for P:

$$\int_\Omega \nabla \cdot (w \lambda \nabla P) - \int_\Omega \lambda \nabla w \cdot (\nabla P) = 0. \tag{2.6}$$

Utilizing the divergence theorem, the first integral in the above equation is transformed into a boundary integral:

$$\int_{\partial\Omega} \lambda (\nabla P \cdot \mathbf{n}) w ds - \int_\Omega \lambda \nabla w \cdot (\nabla P) = 0, \tag{2.7}$$

where $\partial\Omega$ is the boundary of Ω and $\mathbf{n}$ is the unit normal vector to $\partial\Omega$. Because Neumann conditions must be specified along the domain boundary, equation (2.7) is rewritten as:

$$\int_{\Gamma_N} (\hat{v} \cdot \mathbf{n}) w ds - \int_\Omega \lambda \nabla w \cdot (\nabla P) = 0. \tag{2.8}$$

At this stage, P and w are substituted with linear combinations of piecewise interpolants. In the case of IGA, the field approximations for P and w are:

$$P(x,y) = \sum_i^N R_i(x,y) P_i, \tag{2.9}$$

$$w(x,y) = \sum_i^N R_i(x,y) w_i, \tag{2.10}$$

where the concept of unknown variables at nodes (i.e. P_i and w_i) and Lagrange polynomial basis functions are replaced by unknown variables at control points and NURBS basis functions R_i (*see the IGA section for a more thorough discussion on NURBS basis functions and control points*). Regarding w_i, we may choose N instances of w such that

each w_i equals 1 while the others equal 0. The resulting discretization, which may be solved using standard linear solvers, is given in matrix form in eqs. (2.11):

$$K_{IJ}P_J = F_I, \tag{2.11a}$$

$$K_{IJ} = \int_\Omega \lambda \nabla R_I \nabla R_J \, dA, \tag{2.11b}$$

$$F_I = \int_{\Gamma_N} (\hat{v} \cdot \hat{n}) R_I \, dS, \tag{2.11c}$$

where K_{IJ} is the (I,J) entry in the stiffness matrix, F_I is the I^{th} entry in the load vector, R_I is the I^{th} NURBS basis function, P_I is the pressure affiliated with control point I, and Einstein notation is applied. In the sequel, the Neumann boundaries can also include line sink/source interfaces that are internal to the domain.

2.2 Fundamental Equations for Multiphase Study

The second study in this thesis involves the modeling of a one dimensional (1D) water flood displacing oil in an incompressible, homogeneous medium. Included in this section is the selection of a general relative permeability model and a stabilization term for control of the oscillations associated with higher-order modeling of advection dominated flows. Also reviewed is the Buckley-Leverett equations, which provide an analytic solution to the 1D water flood problem and allow for later computation of error.

2.2.1 Two-Phase Flow Equations

The derivation for incompressible, multiphase flow in a 1D homogenous porous medium begins with a mass balance of the situation shown in Figure 2.1:

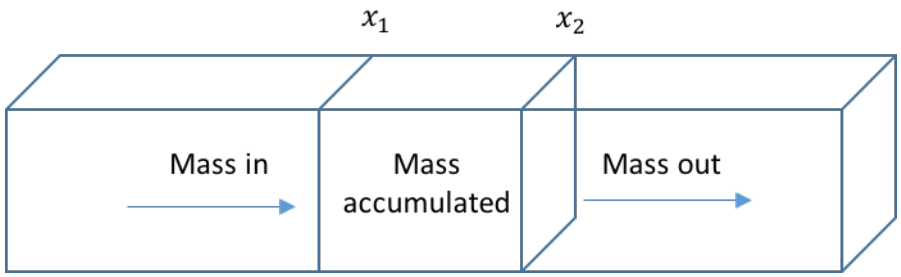


Figure 2.1: Two-phase flow diagram for mass transfer across a representative element in a 1D domain.

$$A\rho_i \cdot (v_i)|_{x_2}^{x_1} = \phi A \rho_i \Delta x \frac{(S_i)|_{t_1}^{t_2}}{\Delta t} \tag{2.12a}$$

$$v_i = v_i(x,t) \tag{2.12b}$$

$$S_i = S_i(x,t) \tag{2.12c}$$

where A is the cross sectional area, ρ_i is the density of phase i, v_i is the Darcy velocity of phase i, ϕ is the porosity, Δx is the length of the control volume, S_i is the saturation (volume fraction) of phase i, and Δt is the time interval. Moving the left hand term to the right hand side, dividing through by volume times density, and taking the limit as Δx and Δt go to zero introduced derivative expressions:

$$0 = \frac{\partial S_i}{\partial t} + \frac{1}{\phi}\frac{\partial v_i}{\partial x}. \tag{2.13}$$

With a general expression relating the saturation and velocity derivatives for each phase, a standard approach would involve substituting Darcy's Law into v_i and rearranging to obtain a non-linear elliptic pressure equation and a non-linear hyperbolic saturation equation. However, for the purposes of this study only the saturation equation is of concern and is represented by:

$$0 = \frac{\partial S_w}{\partial t} + \frac{1}{\phi}\frac{\partial v_w}{\partial x}, \tag{2.14a}$$

$$v_w = \hat{v}_w \quad for \quad v_w \in \Gamma_N, \tag{2.14b}$$

$$S_w = \hat{S}_w \quad for \quad S_w \in \Gamma_D, \tag{2.14c}$$

where the subscript w represents the water saturation, $\hat{v}_w$ is the water velocity on the Nuemann boundaries, and $\hat{S}_w$ is the water saturation on the Dirichlet boundaries.

2.2.2 Inclusion of Stability Terms and Weak Form Discretization

To account for the non physical oscillations arising from higher-order approximations of advection-dominated flow, fine grid phenomena (e.g. saturation shock fronts) must be represented in the coarse scale formulation. To this end, the ASGS method, itself a part of the broader class of VMS methods, is applied to flow in porous media. Procedure for utilization of the ASGS method is carried out according to the works in Juanes and Patzek (2004) and Juanes (2005). First, a non-physical diffusion term D is added within the spatial derivative term of (2.14) to mimic the effects of capillary pressure and eventually apply the stabilization technique:

$$0 = \frac{\partial S_w}{\partial t} + \frac{1}{\phi}\frac{\partial}{\partial x}(v_w - D\frac{\partial S_w}{\partial x}). \tag{2.15}$$

The diffusion term D is further modeled by

$$D = \varepsilon S_w^*(1 - S_w^*), \tag{2.16a}$$

$$S_w^* = \frac{S_w - S_{wirr}}{S_m - S_{wirr}}, \tag{2.16b}$$

where ε is a scaling constant associated with the diffusion model, S_w^* is the normalized saturation, S_{wirr} is the irreducible water saturation, and S_m is the water saturation at irreducible oil saturation. Proceeding in the standard Ritz-Galerkin procedure, a test function w is multiplied to (2.15) and the results integrated:

$$0 = \int_0^L w(\frac{\partial S_w}{\partial t} + \frac{1}{\phi}\frac{\partial}{\partial x}(v_w - D\frac{\partial S_w}{\partial x}))dx. \tag{2.17}$$

Using integration by parts, the previous equation is expanded to include a Nuemann boundary term:

$$0 = \int_0^L \left(w(\frac{\partial S_w}{\partial t}) - \frac{1}{\phi}\frac{\partial w}{\partial x} \cdot (v_w - D\frac{\partial S_w}{\partial x}) \right) dx + \frac{1}{\phi}[w \cdot (v_w - D\frac{\partial S_w}{\partial x})]_0^L. \tag{2.18}$$

The field variables and test function are now approximated by a sum of basis functions defined over the domain:

$$S_w = \sum_{i=1}^{N} R_i \cdot S_{wi}, \tag{2.19a}$$

$$w = \sum_{i=1}^{N} R_i \cdot w_i, \tag{2.19b}$$

where R_i is the i^{th} basis function, and S_{wi} and w_i are the i^{th} nodal values of saturation and test function, respectively. Additionally, time derivatives may be approximated using a backwards difference rule:

$$\frac{\partial S_w}{\partial t} = \frac{S_w^{(n)} - S_w^{(n-1)}}{\Delta t}, \tag{2.20}$$

where the n superscript refers to time step. Plugging in (2.19a) through (2.20) into

(2.18):

$$0=\int_0^L\left(\sum_{i=1}^{N}(R_i\cdot w_i)\frac{\sum_{j=1}^{N}R_j\cdot(S_{wj}^{(n)}-S_{wj}^{(n-1)})}{\Delta t}+\frac{1}{\phi}\frac{\partial(\sum_{i=1}^{N}R_i\cdot w_i)}{\partial x}\cdot(v_w(S_w^{(n+m)})-\right.$$
$$\left.D(S_w^{(n+m)})\frac{\partial\sum_{i=1}^{N}(R_i\cdot w_i)}{\partial x})\right)dx-\frac{1}{\phi}[w\cdot(v_w(S_w^{(n+m)})-D(S_w^{(n+m)})\frac{\partial\sum_{j=1}^{N}(R_j\cdot S_{wj})}{\partial x})]_0^L. \tag{2.21}$$

Note here that v_w and D are defined explicitly as functions of $S_w^{(n+m)}$, with the value of m depending on whether an explicit ($m=-1$) or implicit ($m=0$) scheme is used. Continuing with the discretization procedure, any selection of w_i values must satisfy the above equations. Therefore, the nodal values of w may be chosen N times such that each nodal value is one while the others are zero. This leads to a system of N equations for (2.21). Further rearranging by putting the S_w terms into vector form:

$$0=\int_0^L\left(R_iR_j\cdot\frac{(S_{wj}^{(n)}-S_{wj}^{(n-1)})}{\Delta t}+\frac{1}{\phi}\frac{\partial R_i}{\partial x}\cdot(v_w-D\frac{\partial R_j}{\partial x}S_{wj}^{(n)})\right)-\frac{1}{\phi}[R_i\cdot(v_w-D\frac{\partial R_j}{\partial x}S_{wj}^{(n)})]_0^L. \tag{2.22}$$

Alternatively, in matrix form:

$$0=K_1\cdot\vec{S_w}^{(n)}-K2\cdot\vec{S_w}^{(n-1)}-Q, \tag{2.23}$$

where

$$K_1=\frac{1}{\Delta t}\int_\Omega\begin{bmatrix}R_1R_1&\cdots&R_1R_N\\ \vdots&\ddots&\vdots\\ R_NR_1&\cdots&R_NR_N\end{bmatrix}-\frac{1}{\phi}\int_\Omega D\begin{bmatrix}\frac{\partial R_1}{\partial x}\frac{\partial R_1}{\partial x}&\cdots&\frac{\partial R_1}{\partial x}\frac{\partial R_N}{\partial x}\\ \vdots&\ddots&\vdots\\ \frac{\partial R_N}{\partial x}\frac{\partial R_1}{\partial x}&\cdots&\frac{\partial R_N}{\partial x}\frac{\partial R_N}{\partial x}\end{bmatrix}, \tag{2.24}$$

$$K_2=\frac{1}{\Delta t}\int_\Omega\begin{bmatrix}R_1R_1&\cdots&R_1R_N\\ \vdots&\ddots&\vdots\\ R_NR_1&\cdots&R_NR_N\end{bmatrix}, \tag{2.25}$$

$$Q=\frac{1}{\phi}\int_\Omega\begin{bmatrix}\frac{\partial R_1}{\partial x}v_w\\ \vdots\\ \frac{\partial R_N}{\partial x}v_w\end{bmatrix}-\frac{1}{\phi}\begin{bmatrix}R_1\cdot(v_w-D\frac{\partial R_1}{\partial x})\\ \vdots\\ R_N\cdot(v_w-D\frac{\partial R_1}{\partial x})\end{bmatrix}_0^L, \tag{2.26}$$

$$\vec{S_w}^{(n+m)}=\begin{bmatrix}S_{w1}\\ \vdots\\ S_{wN}\end{bmatrix}^{(n+m)},\quad m=0,-1. \tag{2.27}$$

It is now necessary to establish a stabilization term in order to compensate for the unresolvable fine-scale phenomena. An in depth derivation may be found in Juanes and Patzek (2004), but for conciseness only the main definitions and end results are expressed below. First, a total stabilization term to be added to the right side of (2.22) is designated by

$$\sum_e \int_{\Omega^e} \tau R \mathcal{L}^* w d\Omega, \tag{2.28}$$

where τ is referred to as the intrinsic time, R is the residual associated with (2.17), and $\mathcal{L}^* w$ is the adjoint of the linear advection-diffusion operator. From Codina (1998) and Codina (2000), an effective model for τ under the conditions of linear advection-diffusion type problems and 1st order finite elements is obtained[1]:

$$\tau = (4\frac{|D|}{h^2} + 2\frac{|a|}{h})^{-1}, \tag{2.29}$$

where h is the element length scale and a is an advection term associated with the characteristic saturation equation:

$$a = \frac{\partial v_w}{\partial S_w} - \frac{\partial D}{\partial S_w}\frac{\partial S_w}{\partial x}. \tag{2.30}$$

Additionally, $\mathcal{L}^* w$ is defined as

$$\mathcal{L}^* w = -a \cdot \frac{\partial w}{\partial x} - \frac{\partial}{\partial x} \cdot (D\frac{\partial w}{\partial x}). \tag{2.31}$$

Due to the complexity of (2.28), standard application of the Ritz-Galerkin discretization procedure proves infeasible. Instead, basis functions are substituted into τ, R, and $\mathcal{L}^* w$ individually and matrix-form expressions obtained for multiplication with $S_w^{(n)}$ and $S_w^{(n-1)}$. For clarification, the expressions take the form

$$K_{e,ij} = \int_{\Omega^e} \tau_i R_{ij} (\mathcal{L}^* w)_{ij} d\Omega, \tag{2.32}$$

where $K_{e,ij}$ is the ij component of the e^{th} element stiffness matrix associated with the stabilization term, and τ_i, R_{ij}, and $(\mathcal{L}^* w)_{ij}$ are the respective i or ij entries in the vectors or matrices obtained by substituting in basis functions for S_w and w where applicable. Assembling element matrices into global matrices $K_{VMS,1}$ and $K_{VMS,2}$ gives rise to the final expression which can be used to solve the modified saturation equation:

$$0 = K_1 \cdot \vec{S_w}^{(n)} - K_2 \cdot \vec{S_w}^{(n-1)} - Q + K_{VMS,1} \cdot \vec{S_w}^{(n)} - K_{VMS,2} \cdot \vec{S_w}^{(n-1)} \tag{2.33}$$

[1]The present formulation for τ is also used in the higher order IGA method due to a lack of literature giving more optimal parameters. While the value for h from (2.29) is preserved from the SFEM case, D and a are evaluated under the IGA paradigm when applicable.

2.2.3 Relative Permeability Model

Commonly, power law models based off of the works of Brooks and Corey (Brooks and Corey, 1964) are chosen to represent relative permeability data. The model used in later sections has the form:

$$k_{rw} = k^*_{rw} \cdot (S^*_w)^{\lambda_w}, \tag{2.34}$$

$$k_{ro} = k^*_{ro} \cdot (1 - S^*_w)^{\lambda_o}, \tag{2.35}$$

where k_{ri} is the i^{th} phase relative permeability, k^*_{ri} is the maximum i^{th} phase relative permeability, and λ_i is an exponential fitting parameter associated with phase i. i here is either o for the oil phase or w for the water phase.

2.2.4 Buckley-Leverett Equations

For numerical approximations involving immiscible displacement in 1D media, the analytically derived Buckley-Leverett equations may be used (Buckley and Leverett, 1942). Firstly, making the assumption that capillary and gravity effect are negligible, fractional flow of water is given by

$$f_w = \frac{1}{1 + \frac{\lambda_o}{\lambda_w}}, \tag{2.36}$$

where f_w is the fraction of fluid flow which is water and λ_o and λ_w are the phase mobilities:

$$\lambda_o = \lambda k_{ro}, \tag{2.37}$$

$$\lambda_w = \lambda k_{rw}. \tag{2.38}$$

Next, manipulations of (2.14) lead to a frontal advance equation used to determine the position of a saturation shock front:

$$x = \frac{Q_i(t)}{\phi A} \frac{df_w}{dS_w}, \tag{2.39}$$

where Q_i is the cumulative injection. Referring to Fig 2.2, a graphical approach is often adopted to solve for the shock front characteristics. Plotting the curves for f_w and f'_w, a tangent line is first drawn from the irreducible water saturation to the curve of f_w. The saturation at the intersection, point A, is taken to be the shock front water saturation, S_{wf}. Next, the value of f'_w at S_{wf} is obtained by drawing a line to the right y-axis from point B. This value then goes into (2.39) with other known parameters to determine x_{sf}, the shock front position in the 1D medium. It is noted for clarity that at a given point

in time, all saturations downstream of the shock front remain at initial conditions until those regions are contacted.

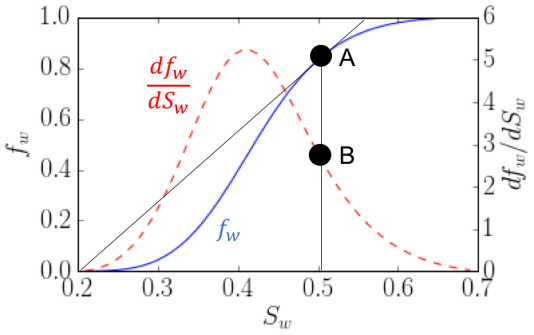


Figure 2.2: Example figure for Buckley-Leverett determination of shock front location.

Chapter 3

Isogeometric Analysis

Below, several fundamental concepts of IGA are addressed. The topics include knot vectors, basis functions, and control nets, which work together to determine the mesh characteristics in subdomains referred to as patches. A thorough introductory discussion of IGA can be found in Cottrell et al. (2009), while useful algorithms for field refinement may be found in Piegl and Tiller (1997).

3.1 Knot Vectors

Knot vectors form the parametric space of the discretization and are comprised of a collection of monotonically increasing, or repeated, points called knots which control the order, continuity, and number of basis functions across their span. They have the form

$$\Xi_i = [\xi_1, \xi_2, ..., \xi_{n+p+1}], \tag{3.1}$$

where Ξ is the knot vector in the parametric ξ direction and each ξ_i is a particular knot.[1] n refers to the number of basis functions in the ξ parametric direction while the p subscript refers to the Bézier order of the basis functions. The first knot is usually taken to be 0, while the last knot typically has a value of 1 but may have any positive value. "Open" knot vectors refer to the case where bases are C^0 continuous at knot vector boundaries, which occurs when the first and last $p+1$ knots are repeated.[2] The multiplicity of interior knots determine the continuity of the basis functions across the patch. For uniform knot vectors, in which the multiplicity of all interior knots is equal to some positive integer m less than or equal to p, bases will have $p-m$ continuity within the patch. Elements, in the context of two- and three-dimensional IGA, are formed under the image of a tensor-product of multiple knot vectors spanning each dimension. In one dimension, elements are simply defined by the knot spans themselves.[3] The

[1] It is convention to use η and ζ for the knots and H and Z for the knot vectors in the 2nd and 3rd parametric direction (if applicable).

[2] For clarity in the notation, basis functions with C^n continuity have n continuous derivatives at the knot associated with the basis function.

[3] Elements can have zero measure in the presence of repeated knots using this definition.

collection of elements given by a single knot vector or the tensor-product of two (in 2D) or three (in 3D) knot vectors is called a patch. Each element will contain a number of bases with support equivalent to

$$b_i = \prod_j^D (p_{ij} + 1), \tag{3.2}$$

where b_i is the number of bases with support over each element in patch i, D is the dimensionality, and p_{ij} is the Bézier order in parametric direction j on patch i. In specifying the value for p, the convention is held where linear functions have order 1, quadratics have order 2, etc.

3.2 B-splines and NURBS

Once a knot vector is defined based on the desired element quantity, continuity between elements, and order of smoothness, the Cox-de-Boor recursive algorithm can be used to determine the B-spline functions that form the basis:

$$N_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.3}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \tag{3.4}$$

where N_i is the i^{th} B-spline basis function evaluated at the parametric coordinate ξ and p goes from 0 to the polynomial order.[4,5] The definition $0/0 = 0$ is used in (3.4). NURBS basis functions are then derived from the B-spline basis functions by projecting the B-spline function to one lower dimensional space by a weighting factor w. As an example, consider the NURBS basis functions for a parameterized two-dimensional domain in parametric coordinates of ξ and η:

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi) M_{j,q}(\eta) w_{i,j}}{\sum_{\hat{i}=1}^{n} \sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi) M_{\hat{j},q}(\eta) w_{\hat{i},\hat{j}}}, \tag{3.5}$$

where q is the polynomial order of the bases in the η direction, m is the number of bases in the η direction, M_j is the j^{th} B-spline function in η, and $R_{i,j}^{p,q}$ is the NURBS basis function affiliated with N_i and M_j having Bézier order p and q in the ξ and η directions, respectively. With the NURBS basis functions, the parameterized geometry–in the case of (3.5) a NURBS surface $\mathbf{S}$–is produced by taking the tensor contraction of the basis

[4] "polynomial" is a misnomer here because the B-splines are rational functions, not polynomials. The convention is a holdover from the Lagrangian polynomials used in traditional finite element analysis.

[5] It is convention to use M and L for defining the B-splines in the remaining parametric directions (if applicable).

functions with a set of control points $\mathbf{B}$:

$$\mathbf{S}(\xi,\eta) = R^{p,q}_{i,j}(\xi,\eta)\mathbf{B}_{i,j}, \tag{3.6}$$

where the (i,j) in $\mathbf{B}_{i,j}$ serves to identify the control point and does not reference of its d components, where d is dimension 2 or 3 in the case of a surface.

3.3 Control Points

Control points are somewhat analogous to nodes in FEA; however, control points are not necessarily co-linear with meshed geometry, but instead provide a lattice within which the mesh is plotted. Within the logic of IGA, control points are organized into a matrix referred to as the control net. The j^{th} column of the control net references the j^{th} physical coordinate of the i^{th} control point in the i^{th} row. A final column is also added to store values for the weight, w_i, of the i^{th} control point. The total number of control points in a given mesh will be equivalent to the total number of basis functions.

3.4 Example Problem

To illustrate the above concepts, the permeable media geometry and boundary conditions in Figure 3.1 are discretized and the resulting pressure fields approximated using IGA. The depicted domain has a k/μ ratio of 10^{-10} $\left[\mathrm{m}^2/\mathrm{Pa{\cdot}s}\right]$.

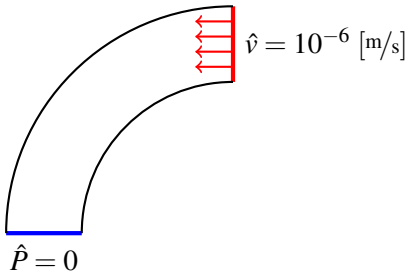


Figure 3.1: Example geometry with boundary conditions.

3.4.1 Initial (Coarse) Mesh

The first step in solving the problem is defining the coarsest set of control points and corresponding knot vectors which accurately capture the geometry. This can normally be accomplished by use of a preferred computer aided design (CAD) software, but here the geometry is simple enough that the control point locations are easily defined. Referencing Cottrell et al. (2009), the control points necessary to define a circular arc are given and lead to the selection of control net and knot vectors in Table 3.1a and

3.1b, respectively. The knot vectors are then used in the Cox-de-Boor formula presented in (3.4) giving the B-spline basis function in the parametric ξ and η directions in Figure 3.2a and Figure 3.2b, respectively. Substituting the B-spline basis functions and control point weights into (3.5) evaluates the NURBS equations that can then be linearly combined according to (3.6) to obtain the mesh (Fig. 3.2c).

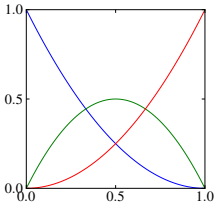
Table 3.1: Example problem control net and knot vectors (coarse).

(a) Control net

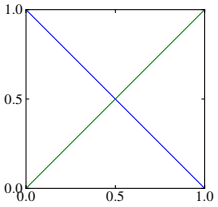
Control Pt.	x	y	weight
1	0	0	1
2	0	1	0.707
3	1	1	1
4	0.707	0	1
5	0.707	1.5	0.707
6	0.707	1.5	1

(b) Knot vectors

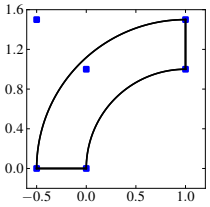
Ξ	$[0,0,0,1,1,1]$
H	$[0,0,1,1]$



(a) ξ basis functions



(b) η basis functions



(c) Coarse mesh

Figure 3.2: Example problem original mesh with associated B-spline functions. Blue squares in (c) represent the control points, and the single patch is highlighted in black.

3.4.2 Refinement

The mesh parameters are now modified to consist of two elements in the parametric directions using both a *h*- and *k*-strategy. The selection of the new control points is not as straightforward as in the coarse mesh and requires application of the algorithms detailed in Piegl and Tiller (1997). The first stage of this process is the selection of the new knot vectors, which is fairly simple. For the *h*-refined set of knot vectors, which simply make use of knot insertion, recall that NURBS bases have inter-element continuity equivalent to their multiplicity minus their polynomial order. Therefore, the addition of another knot with multiplicity 2 is used in *h*-refining the knot vector Ξ while a knot with multiplicity 1 is used in H. The new control nets and knot vectors for the *h*-refined model are summarized in Table 3.2 and the basis functions and mesh is shown in Figure 3.3. For the *k*-refined case, the knot added to Ξ has a multiplicity of 1, raising

the continuity of bases in the ξ direction to C^1. The new control nets and knot vectors are summarized in Table 3.3. The basis functions and mesh are shown in Figure 3.4.

Table 3.2: Example problem control net and knot vectors (h-refined).

(a) Control net

Control Pt.	x	y	weight
1	0	0	1
2	0	0.5	0.854
3	0.25	0.75	0.854
4	0.5	1	0.854
5	1	1	1
6	-0.25	0	1
7	-0.25	0.625	0.854
8	0.0625	0.938	0.854
9	0.375	1.25	0.854
10	1	1.25	1
11	-0.5	0	1
12	-0.5	0.75	0.854
13	-0.125	1.13	0.854
14	0.25	1.5	0.854
15	1	1.5	1

(b) Knot vectors

Ξ	$[0,0,0,0.5,0.5,1,1,1]$
H	$[0,0,0.5,1,1]$

Table 3.3: Example problem control net and knot vectors (k-refined).

(a) Control net

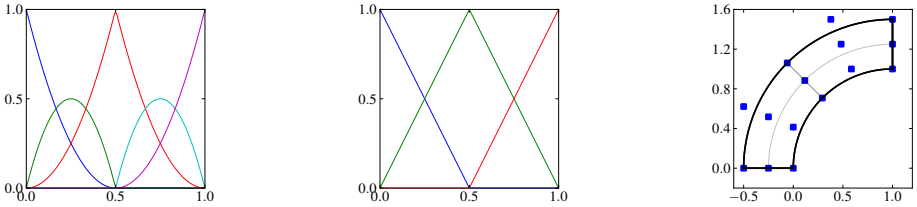
Control Pt.	x	y	weight
1	0	0	1
2	0	0.5	0.854
3	0.5	1	0.854
4	1	1	1
5	-0.25	0	1
6	-0.25	0.625	0.854
7	0.375	1.25	0.854
8	1	1.25	1
9	-0.5	0	1
10	-0.5	0.75	0.854
11	0.25	1.5	0.854
12	1	1.5	1

(b) Knot vectors

Ξ	[0,0,0,0.5,1,1,1]
H	[0,0,0.5,1,1]

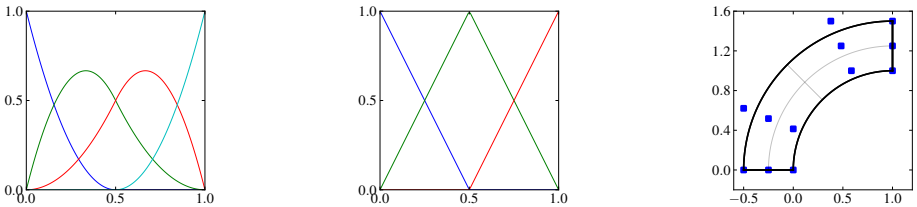
3.4.3 Analysis

A code was developed for analysis utilizing the NURBS functions derived from the B-spline functions and control points in the Ritz-Galerkin matrix structure described in (2.11b) and (2.11c). The resulting approximation for the k-refined case is presented in Figure 3.5. Note that the pressure contours flow naturally with the curvature of the domain, thereby yielding reduced error associated with geometrical mismatch.



(a) ξ basis functions (b) η basis functions (c) Coarse mesh

Figure 3.3: Example problem h-refined mesh with associated B-spline functions. Blue squares in (c) represent the control points, and the single patch is highlighted in black.



(a) ξ basis functions (b) η basis functions (c) Coarse mesh

Figure 3.4: Example problem k-refined mesh with associated B-spline functions. Blue squares in (c) represent the control points, and the single patch is highlighted in black.

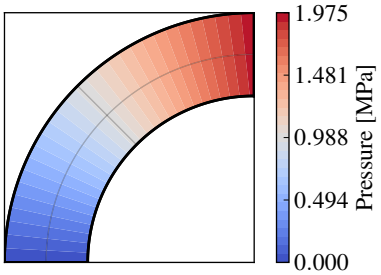


Figure 3.5: Example approximation using the k-refined mesh parameters.

Chapter 4

Procedure and Methodology

4.1 Implemented Softwares

Studies below which involve a direct comparison of IGA with finite difference (FD) methods utilize Computer Modeling Group's (CMG's) IMEX package (FD grid generation and analysis). All refinements carried out with FD use structured grids and a centralized placement of injectors or producers in their respective gridblocks. In the case of comparisons in 2D between IGA and SFEM, Sandia National Laboratories' CUBIT software (finite element method mesh generation) was applied. Unlike FD based refinements, SFEM refinements use unstructured grids that conform to model geometry (as allowed with piecewise polynomials) and a placement of boundary conditions at the corners of subdomains. Lastly, all original code was written using Python with MATLAB fulfilling several auxiliary tasks.

4.2 Refinement Strategies

For all studies in this paper, FD and SFEM undergo $h-$ refinement strategies (uniform reduction in characteristic grid block or element lengths), while IGA refinements include h-, p-, and/or mixed hpk- strategies. All IGA refinements start from the coarsest mesh that captures the geometry of interest, and globally increase either the number of elements within each patch (h-refinement), the NURBS basis function orders within each patch (p-refinement), or some combination of the two with $k-$ refinement (mixed strategy). The coarsest meshes used as starting points for the FD and SFEM refinements were those that contained the lowest resolution through which one could discern the presence of geometrical complexities.

4.3 Verification Studies

Three initial verification studies were carried out to ensure that the code was functioning correctly, as well as to establish a benchmark for evaluating the effect of curvature in the geometrical study.

4.3.1 Laplace Equation

Using the analytic solution to a Laplace problem, convergence rates for IGA are compared for various NURBS orders under an h- refinement scheme. Convergence in IGA should behave similarly to SFEM (Bazilevs et al., 2006), with L2 error related to element length by

$$\|e\|_{L_2} \propto h^{p+1}, \tag{4.1}$$

where $\|e\|_{L_2}$ is the L_2 error, h is the element length, and p is the order of the NURBS functions. Taking the base-10 logarithm of both sides gives

$$log_{10}(\|e\|_{L_2}) \propto (p+1)log_{10}(h). \tag{4.2}$$

Therefore, it is expected that a log-log plot of L_2-error as a function of element length will be a straight line with a slope of $p+1$.

A Laplace equation was defined over a square domain with one corner at the origin and another at $(800, 800)$. The dependent variable was taken to be Φ. Boundary conditions are such that Φ is 0 along the bottom, left, and right boundaries while $\frac{d\Phi}{dy} = \frac{1}{10}\sin(\frac{\pi x}{800})$ along the top boundary. The resulting solution is

$$\Phi(x,y) = \frac{80}{\pi \cosh \pi} \sin\left(\frac{\pi x}{800}\right) \sinh\left(\frac{\pi y}{800}\right). \tag{4.3}$$

4.3.2 Five-spot Well

To further confirm proper convergence behavior, IGA is used to discretize a more relatable reservoir engineering problem. Here, the quarter five-spot well pattern is selected since the boundary conditions and domain simplify quite nicely. A five-spot well pattern is defined in 2D by an infinite set of adjacent squares with injectors at their corners and a producer at their center, with flow assumed to be single phase, incompressible, steady-state and Newtonian and the porous medium to be isotropic and homogeneous. It is noted that the analytic solution for the full five-spot well pattern is given as

$$P(x,y) = \sum_{n=1}^{N} P_i(x,y) = -\frac{\mu}{4\pi k} \sum_{i=1}^{N} q_i \ln[(x-x_i)^2 + (y-y_i)^2], \tag{4.4}$$

where P is the pressure, μ is the viscosity, k is permeability, q_i is the volumetric flow rate at well i, and (x_i, y_i) is the location of well i (Katiyar et al., 2014). Inspection of (4.4) reveals that no flow boundaries exist at the edges of the quarter five-spot in Fig. 4.1, and that injectors and producers may be thought of as point sources/sinks. This, in turn,

causes (2.11c) to simplify to:

$$F_I = q_I \tag{4.5}$$

where q_I is the point source/sink term at the injector/producer located at control point I (because the injector/producers are at the corners of the domain and the domain will be discretized using IGA with open knot vectors, there will be a control point identically at the corners). Additionally, the pressure becomes zero at the two corners where an injector/producer is not located in the quarter five-spot; therefore, we have Dirichlet boundary conditions

$$P(0,L) = P(L,0) = 0,$$

where L refers to the width of the square quarter five-spot.

In the quarter five-spot verification study, reservoir parameters were chosen such that $\mu = 10^{-3}\,[\text{Pa}\cdot\text{s}]$, $k = 10^{-13}\,[\text{m}^2]$, $q_i = \pm 10^{-3}\,\left[\text{m}^3/\text{s}\right]$, and the injector and producer depicted in Figure 4.1 are separated by $400[\text{m}]$ in the x and y directions.

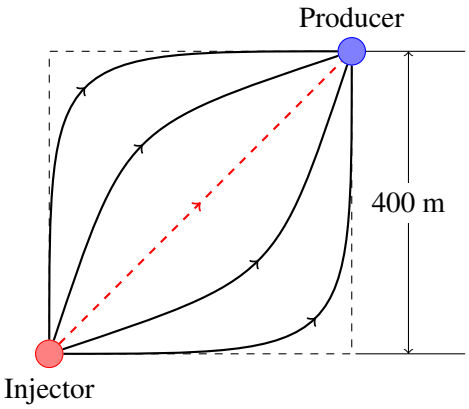


Figure 4.1: Five-spot diagram. The injector is located at the origin.

4.3.3 Simple Line Source

The aforementioned preliminaries verified the IGA implementation for a *single-patch* domain, i.e. a single knot vector in the ξ and η parametric directions was used to discretize the domain. Complex geometries require multiple patches for exact representation. In preparation for an example with a complex geometry that will elucidate the efficacy of the IGA method, verification is performed for a *multi-patch* implementation of a reservoir problem that has a simple enough geometry such that it can also be exactly modeled geometrically with SFEM and FD approaches. A line sink representing an idealized vertical inclusion, e.g. a planar fracture connected to a vertical well or a horizontal well, in the 2D reservoir is now introduced into a well configuration similar to the 5-spot pattern (Fig. 4.2a). Input parameters are $k = 10^{-13}\,[\text{m}^2]$ and $\mu = 10^{-3}\,[\text{Pa}\cdot\text{s}]$. The line sink has a flux of $-5\cdot 10^{-6}\,[\text{m}^2/\text{s}]$. For exterior boundary conditions, pressure

is set to 40 [MPa] at the four corners and volumetric flux is set to zero along the domain edges. No analytic solution is available for this particular problem; a reference solution to the pressure field was obtained using a highly refined IGA mesh and is shown in Figure 4.2b.[1]

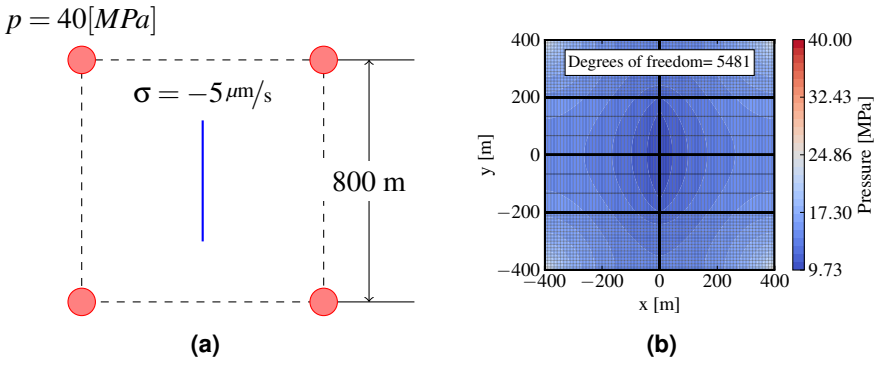


Figure 4.2: Vertical line sink diagram and reference solution. Bold black lines define patch boundaries in (b), while fainter black lines outline element boundaries.

With the defined configuration above, two convergence studies were carried out to compare the IGA approximation against the SFEM and FD approximations. The first study started converging using the lowest global polynomial order that exactly described the geometry (Fig. 4.3a). The coarsest mesh, tailored such that initial element length scale was comparable to that seen in the SFEM method, consisted of eight patches, each with two elements and 1^{st} order bases. The second convergence study treats the initial discretization similarly to the later problem in the complex line sink section. Figure 4.3b depicts a mesh in which the center most patches use 2^{nd} order bases in the parametric direction parallel to the line source. The basis functions for all other patches, as well as for the parametric direction in the center patches that is perpendicular to the line source, are 1^{st} order.

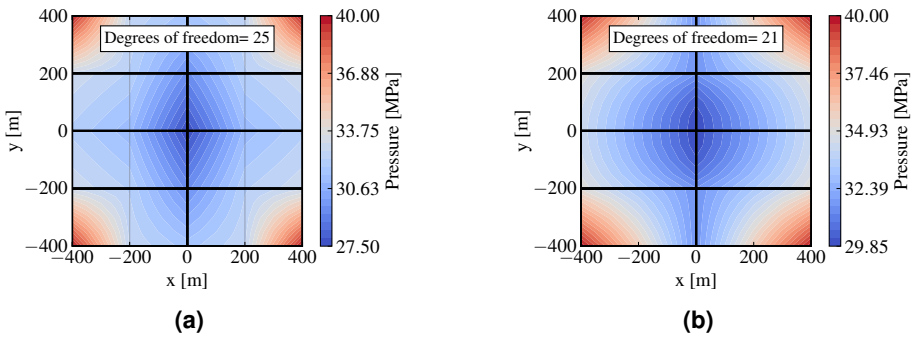


Figure 4.3: Coarsest meshes for vertical line sink verification study. Bold black lines define patch boundaries, while fainter black lines outline element boundaries.

[1]Please see Appendix B for a full list of parameters used to generate the reference solutions.

4.4 Geometrical Study

The geometry from Figure 4.2a was modified such that the resulting line sink could not be exactly discretized by finite elements. A general shape of an *S*-curve was selected based on the known location of control points in IGA for the generation of circular arcs (Cottrell et al., 2009), i.e. there was no need to *actually* use CAD geometry in this case even though that is one of the benefits of IGA. As a means of quantifying the extent to which geometrical complexity impacts the convergence rates of the multiple methods explored, radius of curvature (ROC) of the *S*-curve was varied (Fig. 4.4a, Fig. 4.5a). Boundary conditions are the same as in the vertical line sink problem. Unlike the straight line sink problem, however, the *S*-curve inclusion *requires* the local use of 2$^{\text{nd}}$ order bases for exact representation of the curve. Reference solutions for the two *S*-curves were obtained similarly as before using a highly refined IGA mesh, and can be found in Figure 4.4b and Figure 4.5b.

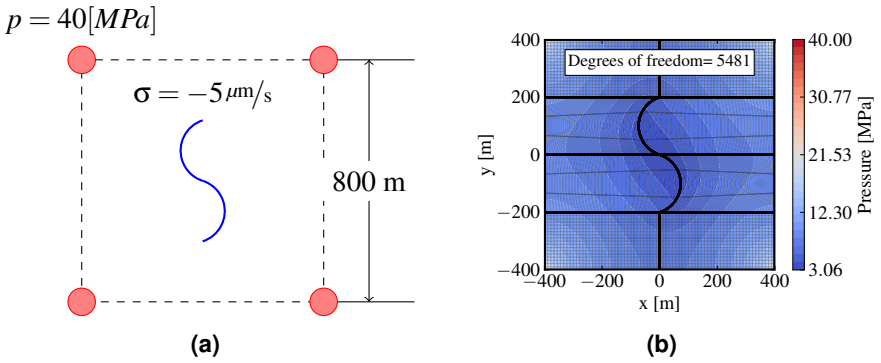


Figure 4.4: *S*-curve line sink diagram and reference solution using a radius of curvature of 105[m].

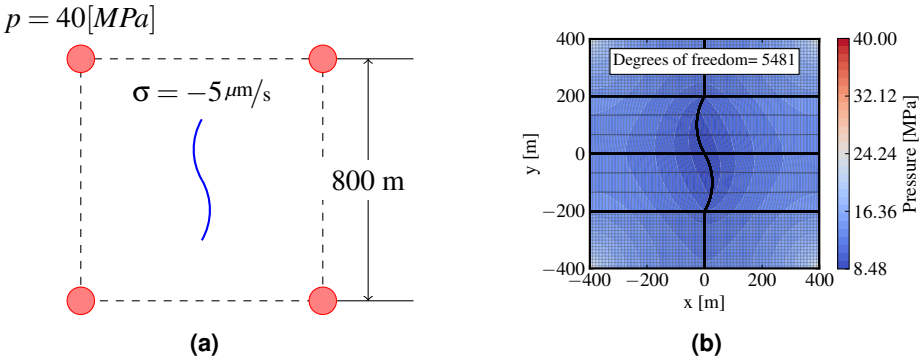


Figure 4.5: *S*-curve line sink diagram and reference solution using a radius of curvature of 200[m].

4.5 Numerical Study

A model is set up to simulate the 1D immiscible displacement of oil by water (Fig 4.6). Physics and geometry are kept fairly simple in order to implement a known analytic solution (Buckley-Leverett) for multiphase flow conditions as well as to easily identify regions or onsets of numerically induced noise. While the ASGS method utilized helps account for steep gradients in the saturation field, it is not a panacea for eliminating error. Rather, it must be determined whether the error that occurs is within acceptable tolerances, especially when compared to the performance of other methods.

Figure 4.6: Diagram of a water flood displacing oil. A saturation front forms at the intersection between a contacted zone and unswept zone.

4.5.1 Paramaters

The physical domain consists of a 1D horizontal reservoir occupied by oil and water. Domain length is $1[m]$, with a water injector at the left hand side set to a constant flux of $1[m/s]$. Initial water saturation is .2. Sensitivity studies are carried out with respect to viscosity ratio ($\mu_r = \mu_o/\mu_w$) to examine the response of error severity. A comprehensive list of fluid and reservoir properties–based off of an example problem from Peters (2012)–is shown in Table 4.1.

Table 4.1: Numerical study parameters.

μ_o	5,10,20 cp
μ_w	1 cp
ϕ	.1
λ_o	3
λ_w	2
k^*_{rw}	.35
k^*_{ro}	.95
S_m	.7
S_{wirr}	.2
ε	.000001

4.5.2 Methodology

Simulations are conducted in a series of $h-$ refinements comparing FD with SFEM and 2nd order (1st order continuous) IGA. Degrees of freedom between numerical approaches are roughly equal, with the SFEM and IGA approaches requiring only one more than the FD method at each level of $h-$ refinement. Results are presented to compare differences in propagating error, characteristics of oscillations (where applicable), and the convergence rate at a representative point in time. L2 error is evaluated using the Buckley-Leverett analytic equations outlined in Chapter 2.

Numerical Approaches

By default, CMG IMEX utilizes an adaptive implicit method (AIM) for evaluation of the saturation equation (Thomas et al., 1983). Conversely, the programmed SFEM and IGA cases are implemented with either a fully implicit or explicit method. For the implicit approach, written algorithms utilize a Newton-Krylov scheme from Python's *scipy* library to solve the large systems of non-linear equations to within a tolerance of 10e-8. Under this regime, the numerical approximation was found to converge only when the VMS terms proposed in (2.33) were neglected (i.e. fine grid phenomena are left unaccounted). While the implicit approach still gives a reasonable estimate of the Buckley-Leverett profile, an additional method was desired which both allowed for implementation of steep-gradient capturing techniques as well as potential computational speed-up. This motivated the use of a modified-explicit routine, where saturation solved for explicitly in each time step is then used to re-evaluate the saturation dependent coefficients. Here, rather than converging to within a certain tolerance, trial-and-error was used to determine the number of iterations past which the method would not yield significant improvement on accuracy.[2] *Please see appendix A for a further description of the workflow as well several illustrative diagrams on the Newton-Krylov and modified-explicit approaches.*

Time Steps

The number of time steps for the higher order cases are determined through the convention

$$\frac{v_{sf} \cdot \Delta t}{\Delta x} = c, \tag{4.6a}$$

$$\Delta t = c\frac{\Delta x}{v_{sf}}, \tag{4.6b}$$

[2]It was determined that with the indicated temporal resolution in the following subsection, 4 iterations of the saturation evaluation within each time step proved sufficient.

where c is a constant, v_{sf} is the shock front velocity, Δt is the time step, and Δx is the distance between nodal points. Designating the number of elements by N_e and shock front breakthrough time by t_{bt}, substitution of L/N_e for Δx and L/t_{bt} for v_{sf} into (4.6b) gives:

$$\Delta t = c\frac{t_{bt}}{N_e}. \tag{4.7}$$

In effect, the number of time steps is chosen to scale linearly with the number of elements. For the iterative scheme described previously, c was selected to be 1 since the method is unconditionally stable and allows for relatively large time steps. For the method which relies on explicit calculations of saturation, c is chosen to be .5 in order to generate stable approximations. For the FD case, the number of time steps in CMG was also selected to be equivalent to the number of elements (i.e. grid blocks). However, CMG uses a dynamic time-stepping that automatically decreases the time step size when convergence of the saturation equation isn't met. In the following chapter, the number of time steps ultimately utilized for each FD refinement are summarized.

Additional Notes

To emphasize, it is not the purpose of this thesis to compare how optimally composed one piece of software is to another, but rather the fundamental suitability of the underlying numerical methods themselves. For this reason, comparisons on the basis of run-time are for the most part neglected. One notable exception occurs later where run speeds between the previously discussed iterative and explicit methods are presented for select cases. Because the explicit approach is somewhat ad hoc and both methods were transcribed using the same programming style and hardware, it was deemed both appropriate and plausible to establish the explicit method's functionality against the well established Newton-Krylov approach.

Chapter 5

Results

5.1 Verification Studies

A series of IGA approximations of (4.3) with C^0 continuous NURBS functions are carried out in Figure 5.1a. Slope of L2 error with element size is seen to increase proportionally with NURBS order. Additionally, a second convergence study is depicted in Figure 5.1b that plots L2 error against number of degrees of freedom and includes C^{p-1} continuous NURBS functions. This second figure shows that on a per degree of freedom basis, more efficient convergence can be achieved by k- refinement where fewer degrees of freedom are needed for similar convergence rates.

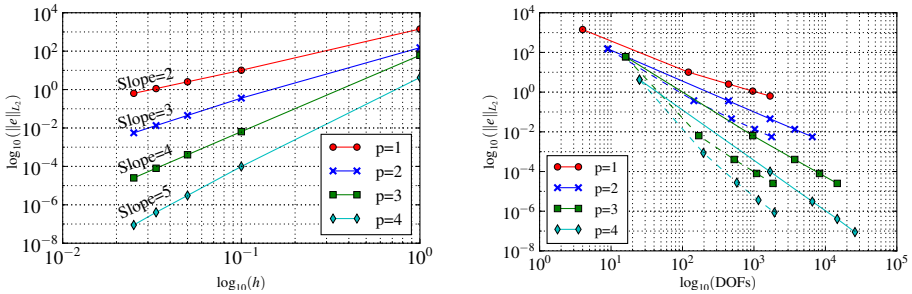


Figure 5.1: Convergence rates for first verification study. (a) depicts C^0 continuous basis function convergence against element length, while (b) depicts both C^0 (solid lines) and C^{p-1} (dashed lines) continuous basis functions against number of degrees of freedom.

Figure 5.2 shows an analytic solution to the 5-spot well problem and an accompanying IGA approximation. The IGA approximation, which has been refined through increasing both element number and polynomial order, is observed to give a close match to the underlying logarithmic equation. As a whole, the pressure profile through the square domain is characterized by steep pressure gradients in the vicinity of injectors or producers and a flatter region in the interstices.

For the two refinement schemes which incorporate a vertical line sink (in place of the producer in an *inverted* 5-spot well pattern), convergence studies are shown in Figure 5.3a and Figure 5.3b . In both cases, the p- refined IGA method has the best

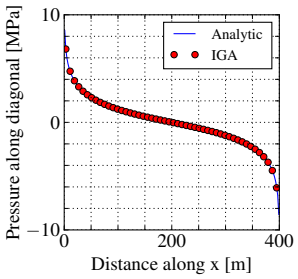


Figure 5.2: Five-spot analytic solution with IGA numerical approximation. The analytic and IGA solution is plotted along the dashed red line in Figure 4.1

convergence rate while the FD method has the worst. In addition, the FD method begins converging from an initial L2 error that is somewhat less than those of the other methods. For Figure 5.3a in particular, the SFEM approximation noticeably overlaps with the *h*- refined IGA solution. In Figure 5.3b, convergence rates are similar overall to those in Figure 5.3a with the exception that the IGA curves have been shifted slightly upward.

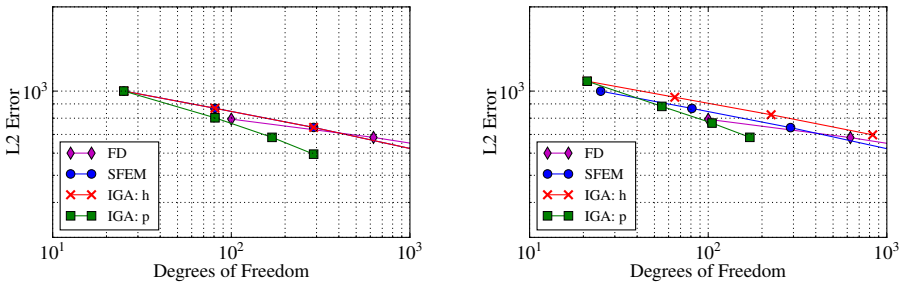


Figure 5.3: Convergence plots for the vertical line sink verification studies.

5.2 Geometrical Study

Replacing the vertical line sink with an $'S'$ curve of varying curvature, convergence plots were obtained for two tested radii of curvature in Figure 5.4a and Figure 5.4b. While slopes of the convergence lines do not change significantly as curvature of the *S*-shaped interface is altered, the various curves shift slightly upward with increasing geometrical dissimilarity from a straight line. Relationships between the effectiveness of the various methods are conserved from the second straight line sink case, with *p*- refined IGA performing more favorably than the SFEM, FD, and *h*- refined IGA methods on a per degree of freedom basis. An additional mixed *hpk*- refinement is presented as well which attempts to focus more refinement in the corners of domain where pressure gradients are steepest (see Appendix B for the parameters used in the *hpk*- refinement).

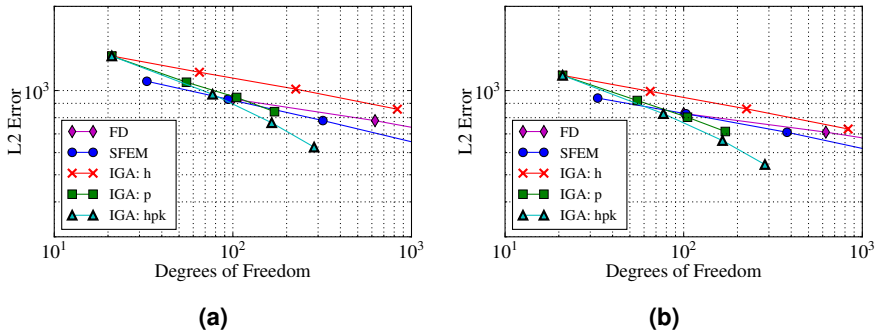


Figure 5.4: Convergence plot for S-curve studies. (a) uses a radius of curvature of 105 [m], while (b) uses a radius of curvature of 200 [m].

5.3 Numerical Study

Analytical breakthrough times and shock front saturations were calculated for each viscosity ratio case (Table 5.1). Based on the proposed time step selection, the results suggest time steps are larger for lower viscosity ratios (while shock front velocity is slower). Additionally, run times of the iterative method and explicit method are compared in Figure 5.5. The graphs show the explicit method runs roughly 1 order of magnitude faster than the its iterative counterpart across all refinement levels and viscosity ratios. Time steps between numerical methods, reported in Table 5.2, show the dynamic nature of the IMEX AIM formulation as well as the previously reported time step selection for the remaining methods.

Table 5.1: Analytic shock front characteristics for various viscosity ratios.

μ_r	$S_w(x_{sf})$	$t_{bt}[days]$
20	.455	.032
10	.500	.036
5	.547	.040

A representative depiction of the shock front propagation is shown in Figure 5.6. Collectively, the graphs depict fairly close agreement of approximate solutions to the analytic solution at coarse scale. For the plots pertaining to the Newton-Krylov based iterative methods, minor oscillations are present behind the shock front which itself is seen to be ”smeared” about the Buckley-Leverett front. Comparatively, the method which incorporates VMS fine grid representation allows slightly better representation of the front steepness, but is also seen to have oscillations down-stream of the shock front. For both cases, where the FD method uses the previously discussed AIM approach, the numerical diffusion effect results in a misleadingly smooth curve at coarse scale which obfuscates the saturation front.

In Figure 5.7, Figure 5.8, and Figure 5.9, curves are shown of the L2 error propaga-

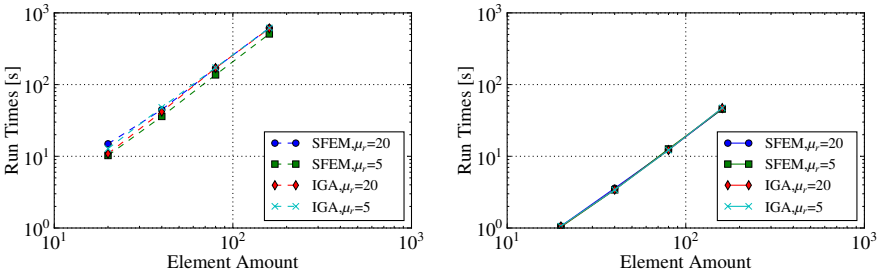


Figure 5.5: Selected comparisons of runtimes between higher order methods at various refinement levels. (a) reflects runs using the implicit method, and (b) reflects runs using the explicit method.

Table 5.2: Time step amounts between methods.

(a) AIM (FD)

μ_r	Elements			
	20	40	80	160
20	21	40	81	174
10	30	54	135	314
5	44	79	159	320

(b) Implicit

μ_r	Elements			
	20	40	80	160
20	20	40	80	160
10	20	40	80	160
5	20	40	80	160

(c) Explicit

μ_r	Elements			
	20	40	80	160
20	40	80	160	320
10	40	80	160	320
5	40	80	160	320

tion for varying viscosity ratios, element numbers, and numerical methods. In general, error is seen to gradually increase with time for all methods and is globally higher when viscosity ratios are higher. Additionally, in the explicit case, seemingly random fluctuations in the L2 curves are both more frequent and more severe with decreasing viscosity ratio. Representative convergencies for the various methods and viscosities, taken when $x_{sf} = .5[m]$, are given in Fig. 5.10. For both numerical schemes, the SFEM and IGA convergence curves essentially overlap, with the IGA curve shifting marginally upward as viscosity ratio decreases in the explicit case. Between the implicit and explicit approach, the latter achieves slightly better convergence rates in addition to curves which occur favorably lower on the plot.

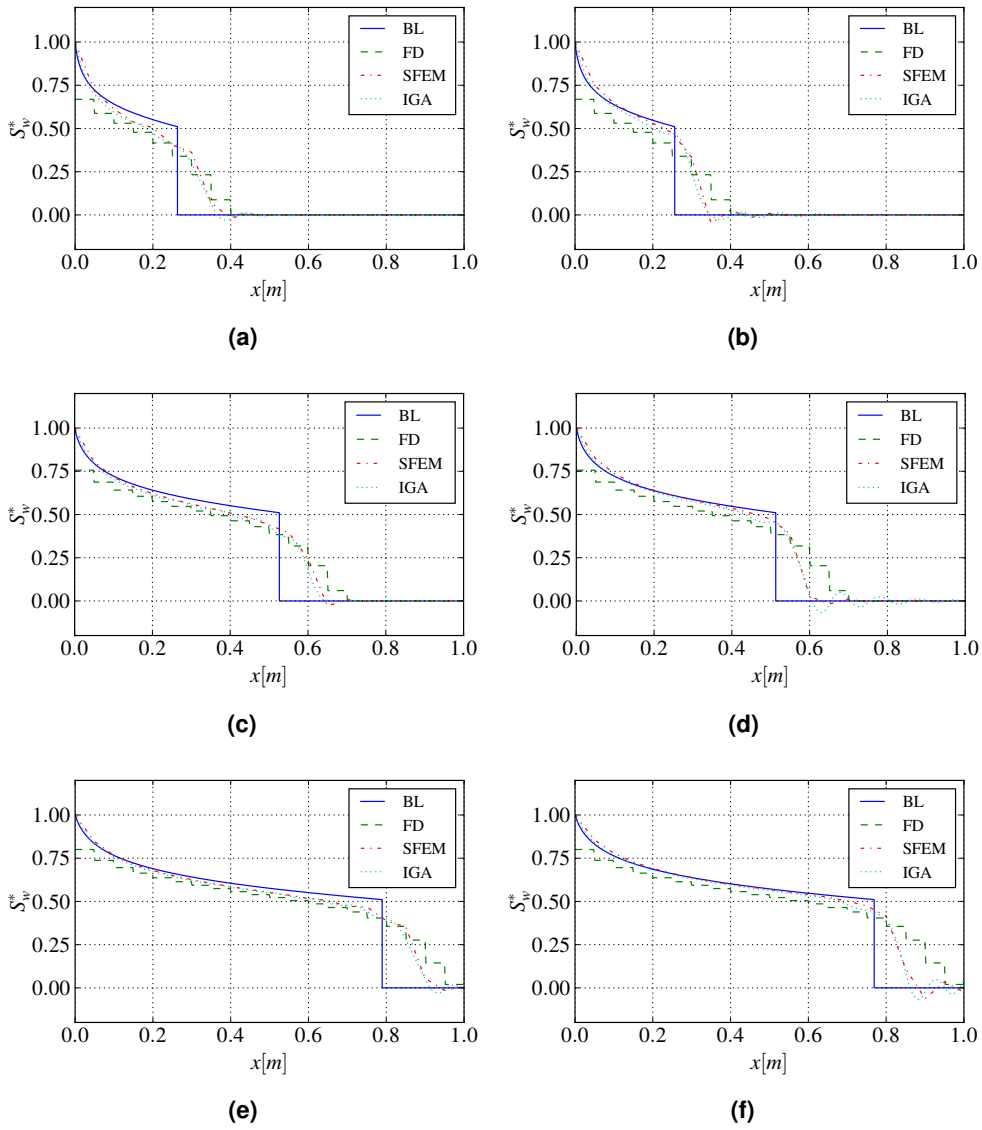


Figure 5.6: Example shock front propagation for the Buckley-Leverett (BL), finite difference (FD), standard finite elements (SFEM), and isogeometric analysis (IGA) cases. Viscosity ratio is 20, and the number of elements is 20. (a),(c), and (e) pertain to the Newton-Krylov based iterative scheme, while (b),(d), and (f) pertain to the explicit scheme.

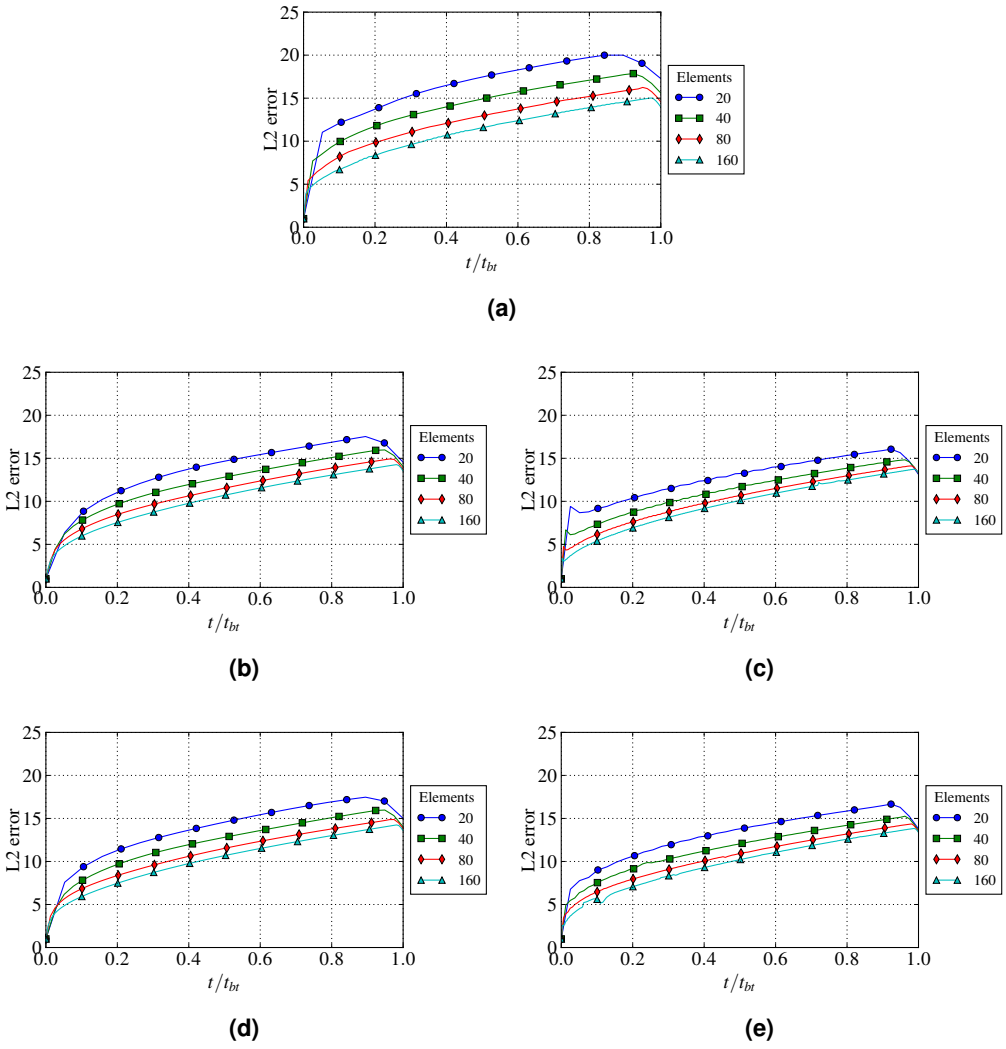


Figure 5.7: L2 error propagations for FD, SFEM, and IGA method. (a),(c), and (e) pertain to the Newton-Krylov based iterative scheme, while (b),(d), and (f) pertain to the explicit scheme. Viscosity ratio is 20.

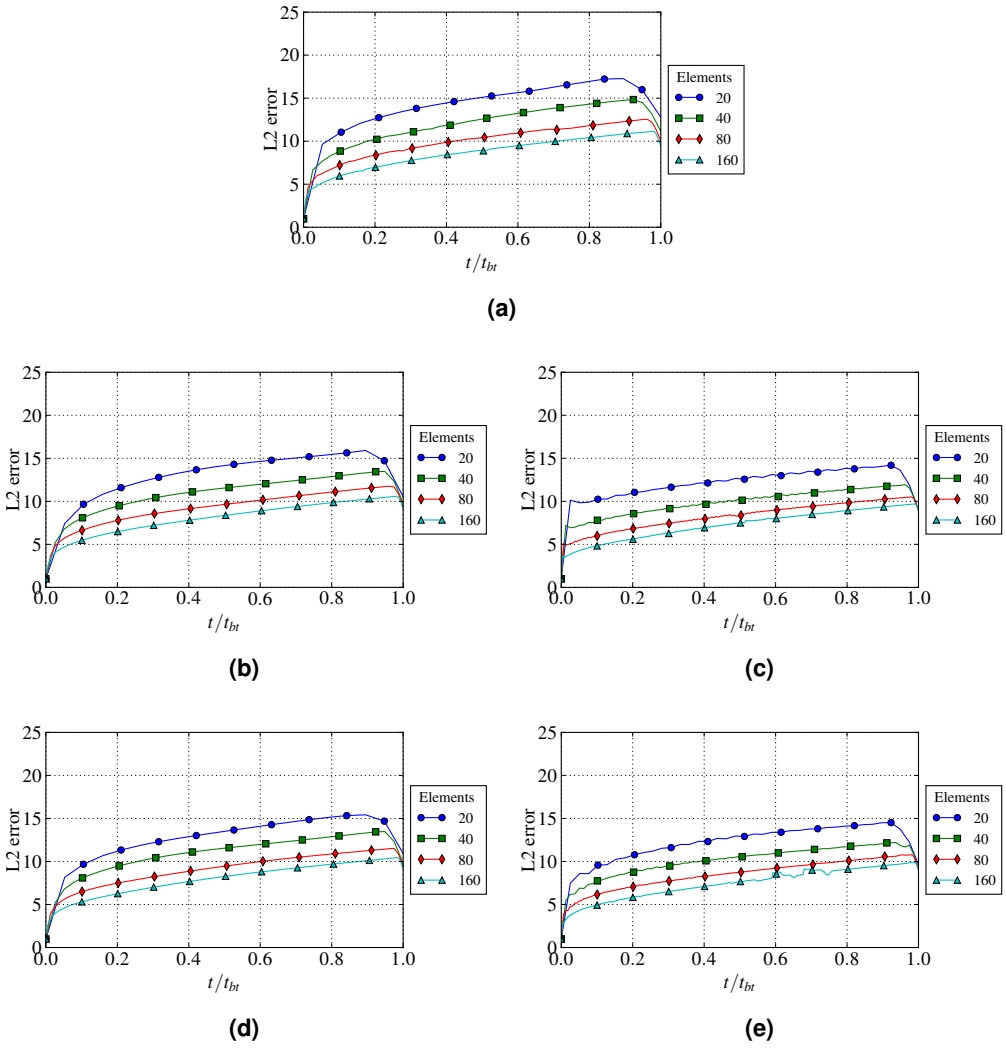


Figure 5.8: L2 error propagations for FD, SFEM, and IGA method. (a),(c), and (e) pertain to the Newton-Krylov based iterative scheme, while (b),(d), and (f) pertain to the explicit scheme. Viscosity ratio is 10.

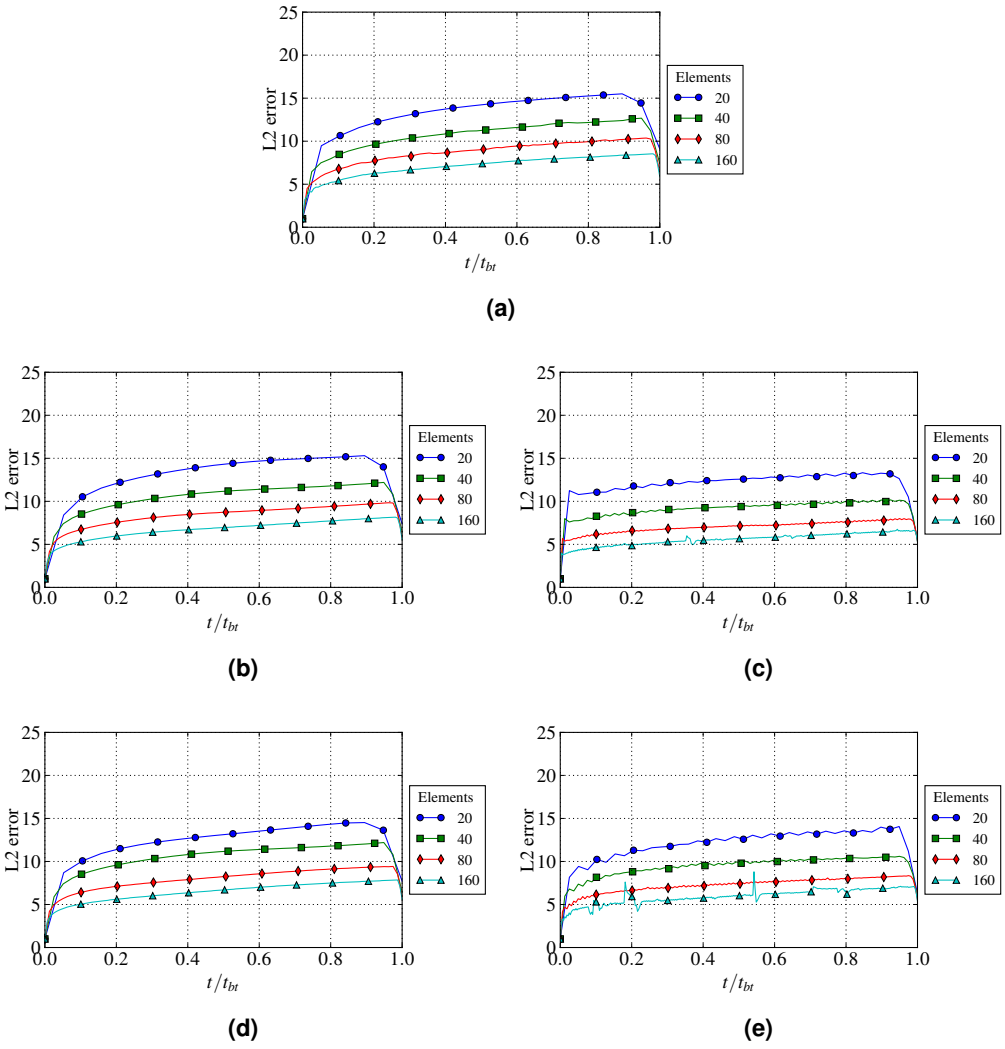


Figure 5.9: L2 error propagations for FD, SFEM, and IGA method. (a),(c), and (e) pertain to the Newton-Krylov based iterative scheme, while (b),(d), and (f) pertain to the explicit scheme. Viscosity ratio is 5.

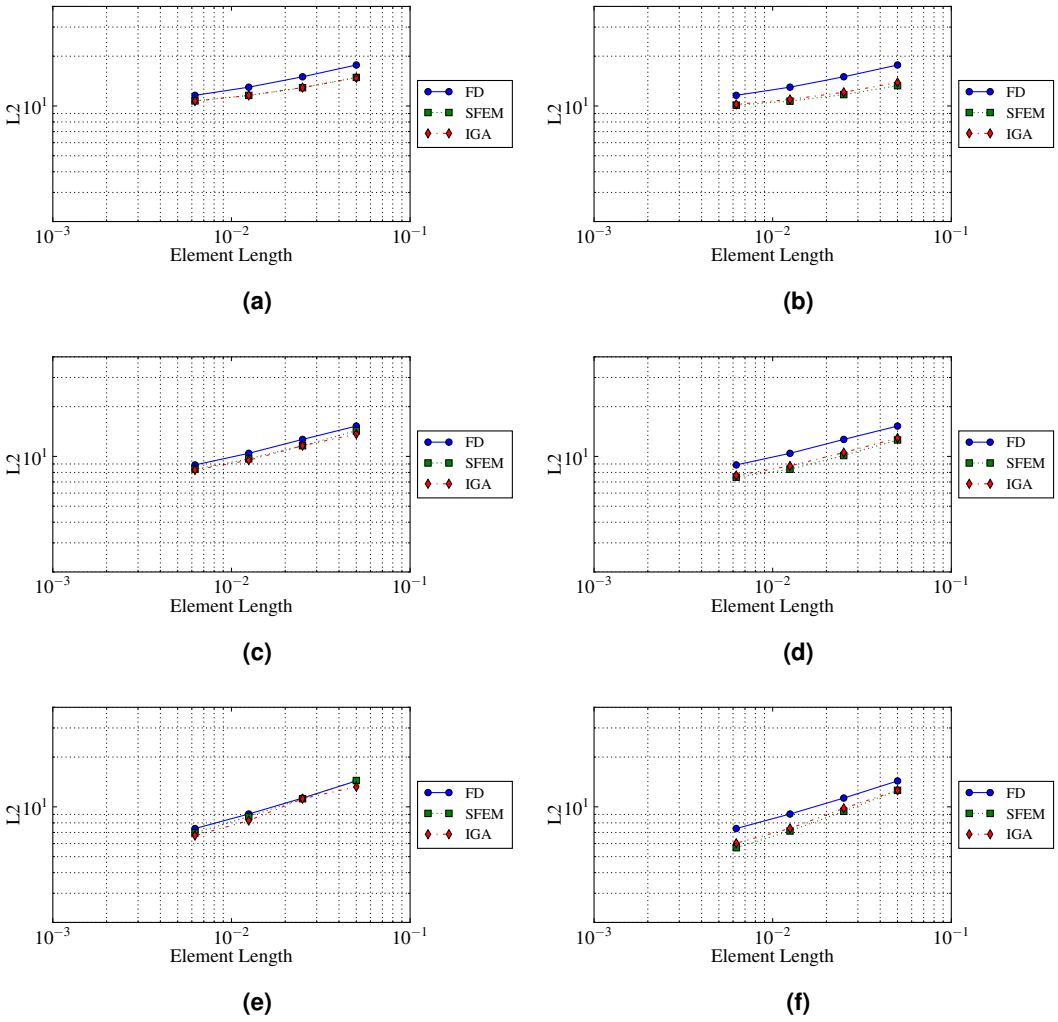


Figure 5.10: L2 convergence for various methods across tested viscosity ratios. The first and second columns apply to the iterative and explicit approaches, respectively. The first, second, and third rows apply to viscosity ratios of 20, 10, and 5, respectively.

Chapter 6

Discussion

6.1 Verification Studies

Results from the Laplace and 5-spot verification studies demonstrate that the IGA code operates correctly in a domain free of geometrical inclusions. The IGA approximation in each case matched the behaviors predicted by (4.3) and (4.4) while respectively showing the potential benefit of $k-$ refinement and the ability of IGA to accurately model steep pressure gradients given a sufficiently refined mesh. In the corresponding studies with the vertical line sink, IGA's capturing of steep pressure gradients is reiterated in a manner that allows closer comparison with the geometrical studies.

Comparing the plots in Figure 5.3a and Figure 5.3b, the local application of 2^{nd} order bases has made the IGA convergence overall less efficient. It can be inferred from Figure 4.2b that the approximation is largely determined by the resolution at the corners, where the pressure gradients are steepest–as indicated by the dense packing of contour lines–and therefore require high h- or p- refinement to model. It follows that adding additional degrees of freedom in patches not containing the corners will lead to relatively small reductions in error, which accounts for the performance discrepancy between the two convergence studies. In essence, although IGA is more flexible with allowing basis functions of varying order to be used throughout the domain, it is not always an efficient strategy to do so.

The results from the convergence study in Figure 5.3a suggest that the IGA code functions correctly when including a line sink into the discretization of a quasi-five-spot well pattern, since the h- refined IGA approximation expectedly overlaps with the SFEM approximation (SFEM and IGA have equivalent basis functions in the case of 1^{st} order). In addition, the appearance of FD to begin converging from a lower L2 error than the other methods is thought to be coincidental with regard to meshing strategy and so is not a proper indicator of the method's performance; therefore, the slope of the convergence is given higher importance than the y-intercept when determining the relative efficiency of the various methods and refinement strategies. Given the correct matching of the IGA solution to the generic partial differential equation and five-spot analytic solution, as well as the correct convergence behavior when subjected to a sim-

ple line sink, the versatility of IGA can be further investigated by applying a line sink for which no exact discretization can be made using the traditional finite difference or finite-element approaches.

6.2 Geometrical Study

To account for the similarity in convergence slopes of the straight and complex line sink cases, note the similarity between Figure 4.4b, Figure 4.5b, and Figure 4.2b. A similar pressure profile between the reference solutions–essentially a flat region at the domain center and steep gradients at the corners–indicates that using consistent mesh and refinement properties between cases should result in correspondingly consistent L2 errors. This again highlights the need to capture the steep pressure gradients at the corners, which in the case of curved geometries play heavily into the optimal selection of meshing strategy. To this end, the tested *hpk*- strategy proved effective in offering a robust placement of higher order NURBS functions and smaller elements in regions where they are most needed (with the added benefit of requiring fewer degrees of freedom due to *k*- refinement). The improved convergence rate over the *p*- refined IGA approach motivates the need for future studies on optimally selecting control point placement and IGA mesh parameters based on the *a priori* evaluation of pressure fields and boundary conditions.

Also worth mention are the initial discretization for the FD, SFEM, and IGA methods. With the structured mesh used in the FD case (Fig. 6.1a, Fig. 6.2a), it is difficult to make out the true geometry being discretized due to the coarseness involved at 100 degrees of freedom. The SFEM improves on this by using a meshing strategy that conforms to the curves (Fig. 6.1b, Fig. 6.2b), but the restriction to straight edged elements still obfuscates the curves of interest. Conversely, the IGA meshes, for roughly $2/3$ the degrees of the SFEM method and $1/5$ the degrees of freedom of the FD method, exactly capture the curvatures of interest (Fig. 6.1c, Fig. 6.2c). The precision exhibited by IGA is a direct result of the IGA basis function structure, which, as explained previously, incorporates the functions used to generate conic sections in CAD.

Even for these basic reservoir configurations, where only one line curve is considered, physics considerations are vastly reduced, and where boundary condition selection partially mask the effect of complex geometries, notable improvements in computational efficiency were made by implementing IGA over the FD and 1^{st} order SFEM method. In particular, the number of degrees of freedom expected for FD or SFEM to resolve features would only be exacerbated under realistic conditions where, for example, networks of fractures exist on multiple scales or where multiple deviated wells are drilled to produce from a reservoir.

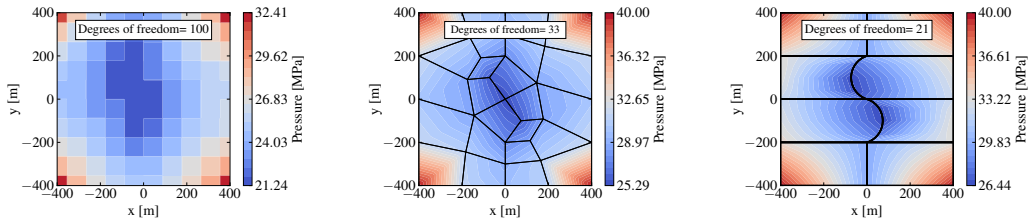


Figure 6.1: Coarsest meshes for S-curve with ROC=105m. Thin black lines define SFEM element boundaries in (b), while fainter black lines and bold black lines define IGA element and patch boundaries in (c), respectively.

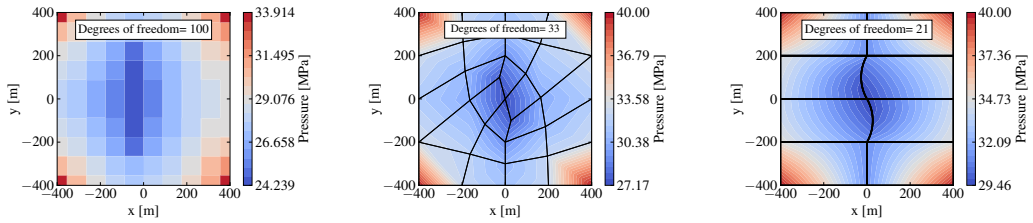


Figure 6.2: Coarsest meshes for S-curve with ROC=200m. Thin black lines define SFEM element boundaries in (b), while fainter black lines and bold black lines define IGA element and patch boundaries in (c), respectively.

6.3 Numerical Study

6.3.1 Time Steps

Behavior of the IMEX dynamic time stepping in Table 5.2 may possibly be attributed to the characteristics of the fractional flow curve. As is in the case of IMEX, Newton's method is a popular choice of iterative solver for nonlinear equations. However, issues arise when attempting to solve sigmoidal functions. Given an initial guess which is insufficiently close to the current time step's solution, Newton's method may not necessarily converge (Jenny et al., 2009; Cogswell and Szulczewski, 2017). This issue is exacerbated for functions with smaller slopes, such as with fractional flow curves which incorporate lower oil viscosities (Fig 6.3). As a remedy, time step size may be dynamically reduced such that the initial guess–the default of which in IMEX is the previous time step's solution– increasingly resembles the current time step solution when needed. The above is evidenced in the presented data, where time steps utilized by IMEX increase with decreasing oil viscosity. For the remaining methods, selection of times steps has proved to be sufficient by virtue of the successful convergencies.

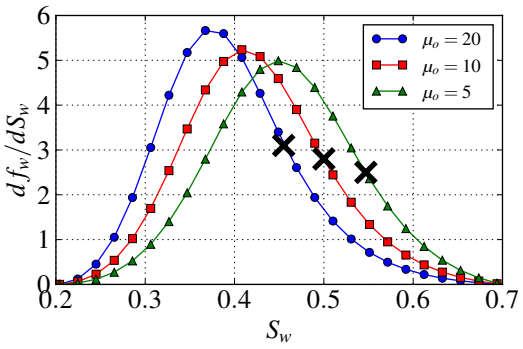


Figure 6.3: Derivative of fractional flow curve w.r.t. S_w for various oil viscosities. Black x's indicate shock front saturation.

6.3.2 Implicit vs. Explicit

Differences in run-times between the NK-implicit and modified-explicit approach are due to the conditions upon which the iterations within each time step will terminate. Although the implicit approach employs fewer time steps, many iterations of the Newton-Krylov method must occur to achieve the desired tolerance (increasingly so for higher degrees of freedom). The explicit method, on the other hand, proceeds to subsequent time steps after a predefined, small number of iterations. The speed issues with the generic implicit method are typically alleviated by careful selection of a preconditioner (Knoll and Keyes, 2004), but this goes beyond the scope of the current study.

Apart from run-time, adopting the explicit method was also advantageous on an error vs. degree of freedom basis. Although the methods were both seen to give good approximations to the analytic solution in Figure 5.6, the explicit method was able to capture the upstream side of the shock more accurately due to implementation of the ASGS equations. Viewing another example of the BL shock front (Fig. 6.4), it is also clear that the approximation obtained with the explicit approach descends more rapidly after the shock front and thus gives a better representation of front steepness. While the pronounced downstream oscillations in the same method ultimately introduce additional error, it was not enough to overcome the contribution from the front capturing (Fig. 5.10).

6.3.3 Comparisons Between FD, SFEM, and IGA

For the general trends in L2 error propagation, the characteristic rise and sudden drop off at shock front breakthrough of the L2 error results from the quality of the early and late time approximations. For all methods, L2 error calculations immediately after the start of injection are dominated by downstream saturation modeling. In this region, the methods exactly capture the analytic solution since the shock front has not yet advanced and initial conditions remain intact. However, as time progresses the L2 error is influ-

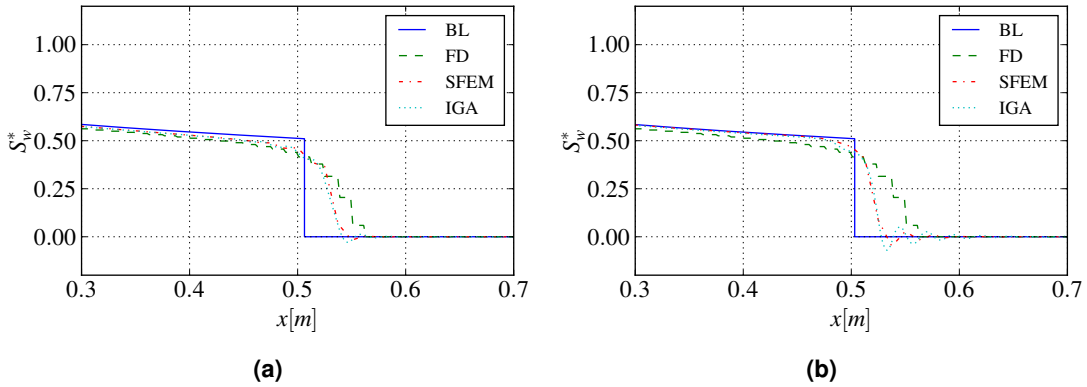


Figure 6.4: Example of near shock approximations. For both figures, μ_r is 20, number of elements is 80, and the shock front is 1/2 the way to breakthrough. (a) gives the NK implicit approach and (b) gives the modified explicit approach.

enced more heavily by upstream modeling and near front phenomena for which the various methods are not exact. Close to breakthrough, as "smeared" or malformed regions immediately downstream of the shock-front contact the outlet of the domain, numerical artifacts are resolved and the error decreases sharply. However, since the methods are inexactly capturing the domain–what is now essentially one large upstream shock front zone–the L2 error does not return to initial values.

The overall higher error associated with FD is to be expected as the method relies on piecewise constants. Within the implicit and explicit method, the somewhat identical performance between SFEM and IGA cannot be explained as easily. While both methods are more accurate than FD by virtue of their higher order basis functions, there is an indiscernible trade off between NURBS order, pre-front oscillations, and treatment of τ and h (as described in footnote 1 in Chapter 2). For the explicit method in particular, the evaluation of coefficients at intermediate approximations to the saturation vectors of each time step may also contribute to the seemingly random fluctuations at lower viscosity ratios and higher element numbers (Figure 5.8). Despite the unpredictability though, eventual convergence to the Buckley-Leverett solution was achieved and the two methods were demonstrated to function. Future studies should work to determine the robustness of these approaches as well as the optimal VMS parameters for higher order NURBS functions.

Chapter 7

Conclusion

This two-part exploration into IGA's use as a future tool for reservoir simulation demonstrates its relative effectiveness to more commonly used methods in the context of both geometrical exactness and numerical resolution. Although the importance of geometrical discretization is somewhat reduced in the complex line-sink study due to the more pressing need of capturing the pressure gradients in corners far away from the *S*-curve, IGA is still seen to outperform SFEM and FD on multiple fronts. Geometrically, IGA was able to exactly discretize the idealized, complex reservoir feature with relatively few degrees of freedom while maintaining said precision during subsequent refinement. Additionally, *p*- and *hpk*- refined IGA was observed to outperform SFEM and FD numerically, with *hpk*- offering a robust tool for optimally selecting control point placement based on the *a priori* evaluation of pressure fields and boundary conditions. For the Buckley-Leverett study, IGA improved upon FD and rivaled SFEM using both a traditional implicit approach as well as the *ad hoc* explicit approach. The explicit approach, as well the more optimal selection of ASGS parameters for higher order IGA methods, may constitute future research topics in and of themselves. Overall, the avenues of research into IGA's application to reservoir simulation are vast and it's outlook promising as a successor or compliment to more industry standard methods .

Appendix A

Code Architecture and Workflows

The following sections give a general summary of the code organization as used in the original IGA programs. Notable resources where the reader may further explore recommended coding practices and underlying algorithms are provided throughout.

A.1 Preliminary Computation of Data Arrays and Control Nets

A.1.1 Patch-data Objects

Data arrays of knot vectors and basis function connectivity must be generated before analysis can begin of the desired equations. User-defined parameters for mesh properties in each patch (i.e. NURBS order, number of elements, and multiplicities), along with information for how the patches are globally connected are sufficient for determining the outputs necessary for later assembly of the global stiffness matrices. In Figure A.1, a diagram is presented for a Python class which takes basic mesh data as an input, computes the necessary arrays, and stores said arrays in readily available lists. While the construction of knot vectors is outlined in Chapter 3, details of the connectivity arrays unique to IGA may be found in the appendices of Cottrell et al. (2009).

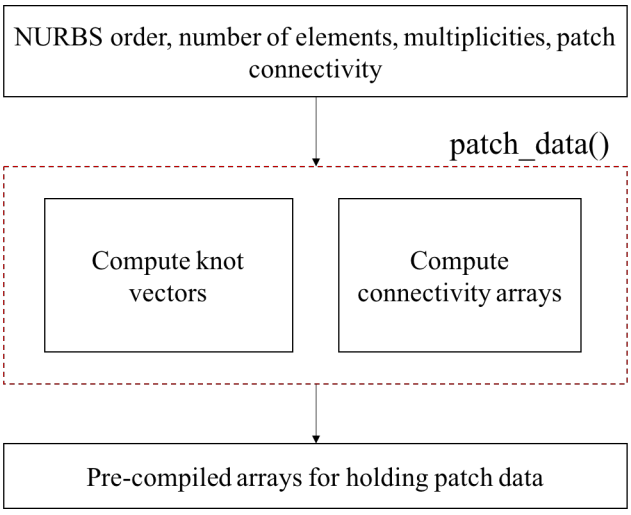


Figure A.1: Flow chart for initial construction of patch data arrays.

A.1.2 Refinement

Automatic refinement routines accept a coarse-scale control net, coarse-scale *patch_data()* object, and refined-scale *patch_data()* object as inputs and return a corresponding refined-scale control net (Figure A.2). At the lowest level, $h-$, $p-$, and/or $k-$ refinement occurs along a single parametric direction in a single patch. Once the changes are made and stored, refinement then takes place in any additional parametric directions that may be present. At the outermost level, this process is then repeated over each patch until the entire domain is refined. As mentioned previously in this thesis, specific details of the algorithms used for $h-$,$p-$, and $k-$ refinement may be found in Piegl and Tiller (1997).

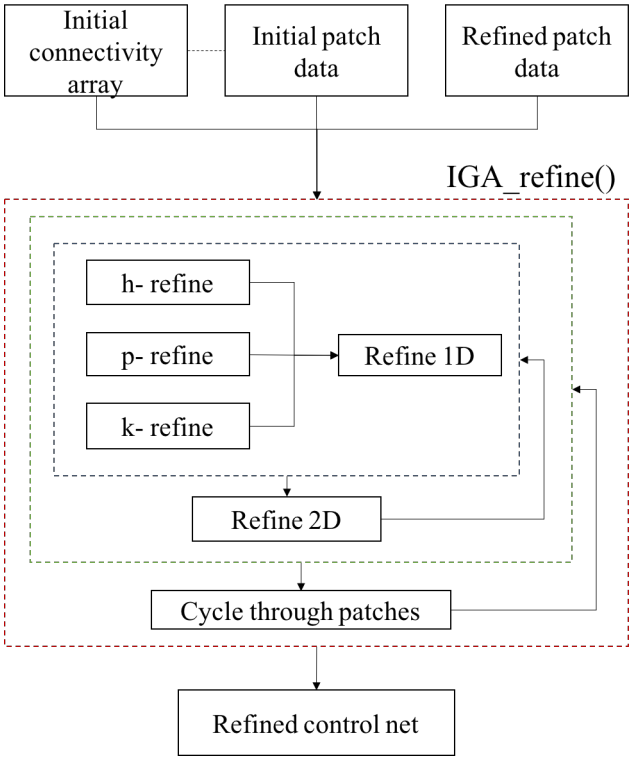


Figure A.2: Flow chart for the refinement routine. Note that 2D refinement is the highest dimension shown since 3D refinement was not of concern in the presented studies.

A.2 Analysis

A.2.1 Format for 2D, Steady-state Study

Analysis of the 2D reservoir with steady-state pressure fields incorporates output from the aforementioned *patch_data()* class along with an associated control net. In addition, reservoir properties must be defined to evaluate coefficients in the global stiffness matrix and load vector. The reservoir properties entered include the following: permeability, viscosity, porosity, and boundary condition locations/values.

In Figure A.3, the general analysis procedure is outlined. First, the location of all gauss integration points are determined such that all necessary mappings from physical to parent-element space and basis function evaluations can be completed simultaneously. Once the mappings and NURBS functions are found, the discretization matrices/vectors are assembled and subsequently updated to reflect boundary conditions. "Nodal" values for pressure are then easily obtainable through simple matrix inversion. The end result of the process uses the discrete pressure values to define a function that returns pressure anywhere in the reservoir given a patch number and parametric coordinate.

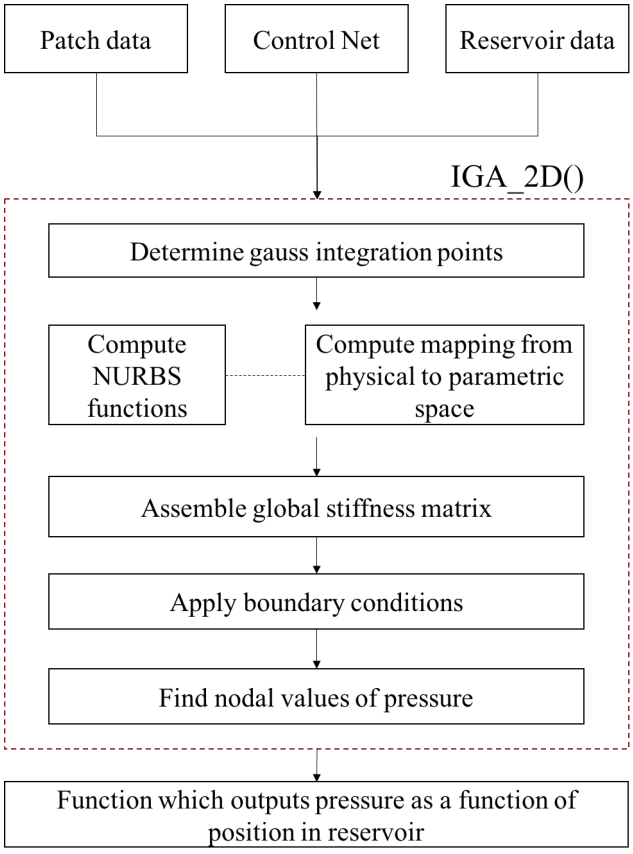


Figure A.3: Flow chart for IGA analysis in 'S' curve study.

A.2.2 Format for 1D, Time-dependent Study

The general structure for the 1D reservoir analysis mirrors that in the 2D case; however, modifications are made in order to handle the time dependency introduced by the saturation equation. In Figure A.4, a new input includes a time step array that is used to evaluate the Δt term in (2.20). Additionally, the process for assembling the needed matrices and vectors includes extra steps due to the iterations in time. These are contained within the dotted green box in Figure A.4, and follow from the choice of explicit or implicit method. The two methods utilized in the 1D study are explained further in

the following subsections.

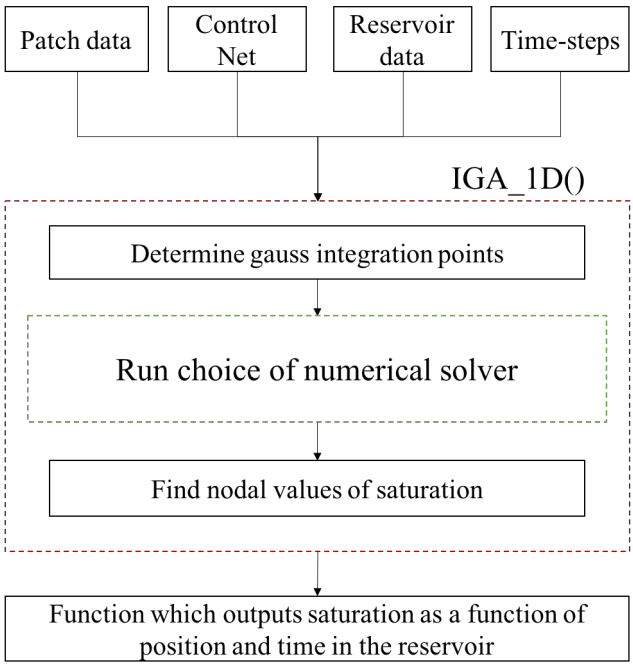


Figure A.4: Flow chart for initial construction of patch data arrays.

Implicit Method

The implicit approach begins with the evaluation of the NURBS functions and coordinate mappings at the integration points.[1] The Newton-Krylov function inherent to Python is then called to evaluate the residual resulting from the stiffness matrix, load vector, and "guess" for saturation in (2.23). If the residual is found to be within a certain tolerance, the time step values for saturation are saved and used as the initial guess in the next time step. If the tolerance is not reached, the Newton-Krylov method continues to search for the root until a valid vector for saturation is acquired. Once the final time step is completed, a function is created that accepts an x-coordinate and time step as inputs and returns a corresponding saturation in the reservoir.

Explicit Method

Rather than computing the residual, the explicit approach finds trial values for saturation by matrix inversion of (2.33):

$$\vec{S_w}^{(n)} = K_1^{-1} \cdot [K_2 \cdot \vec{S_w}^{(n-1)} + Q - K_{VMS,1} \cdot \vec{S_w}^{(n)} + K_{VMS,2} \cdot \vec{S_w}^{(n-1)}]. \tag{A.1}$$

[1] Because the mesh parameterization does not change in time, the NURBS functions and spatial mappings are constant for each temporal iteration.

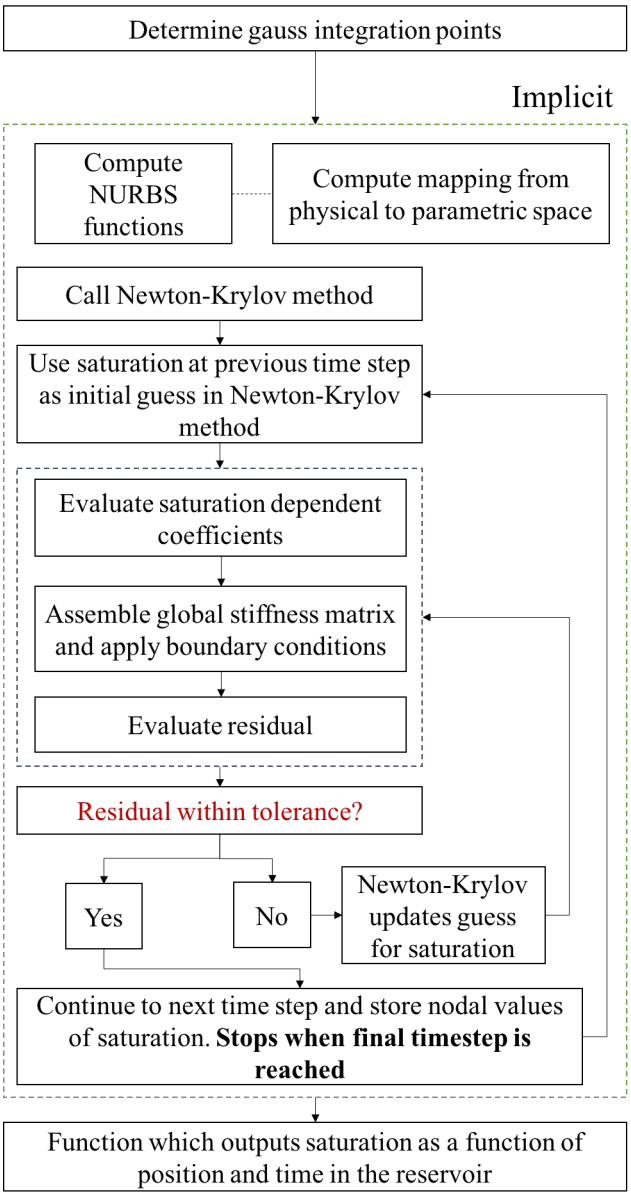


Figure A.5: Flow chart for initial construction of patch data arrays.

For a given number of iterations, the saturation found in this fashions are then used to reevaluate the coefficients and subsequently resolve the problem (Figure A.6). Once the specified number of iterations is reached for finding a time step's saturation distribution, the process is repeated for the remaining time steps. The end result of the explicit approach, similar to that of the implicit method, is a function which can be used with the Buckley-Leverett solution to evaluate the L2 error.

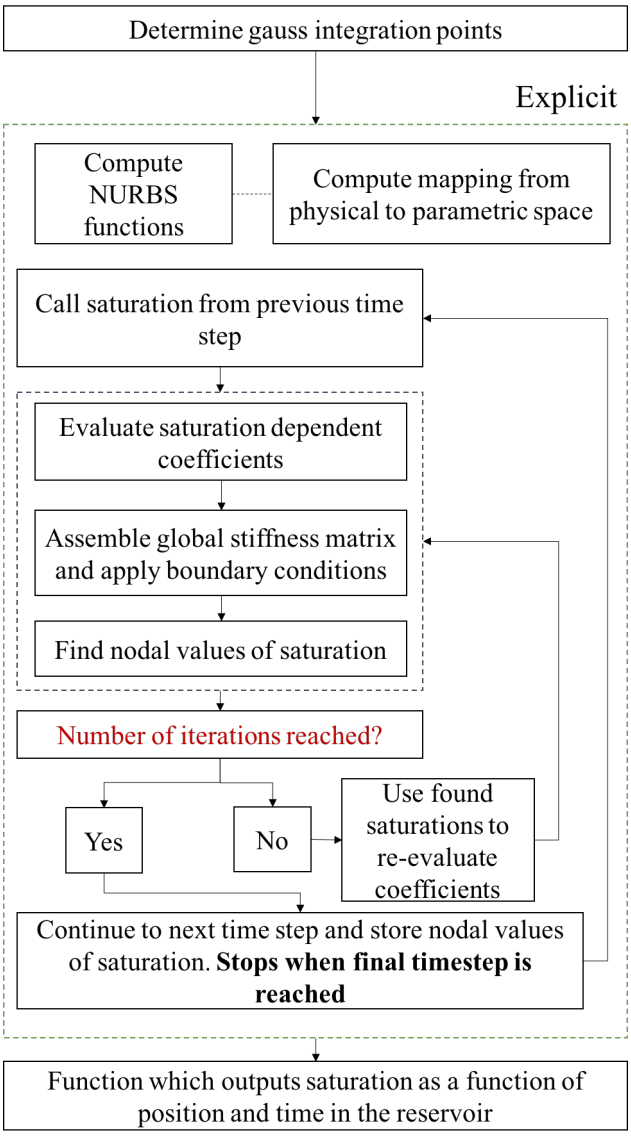


Figure A.6: Flow chart for initial construction of patch data arrays.

Appendix B

High-level Refinement Parameters

B.1 *hpk*- parameters

The 2D studies which incorporate the centralized reservoir feature used separate h, p , and mixed *hpk*- refinement strategies. While the h and p refinement cases employ globally uniform enhancements to either element size or NURBS order, the *hpk*- strategy is not so straightforward. Below, tables are presented which detail the mesh characteristics within each patch for each refinement carried out under the *hpk*- paradigm. Tables are split into the three categories which define mesh characteristics in each patch: knot multiplicity (basis continuity), NURBS order, and number of elements. As presented in the tables, three refinement (along with the initial coarse mesh) are associated with an adjacent data set. Each bracketed pair of values in the data set refers to a ξ and η designation of the indicated mesh property within a particular patch. The position of the bracketed pair within the data set, anywhere in the range of 1-8, refers to the patch across which those parameters are applied. Figure B.1 gives the numbering of patches and should be referenced when reading the tables.

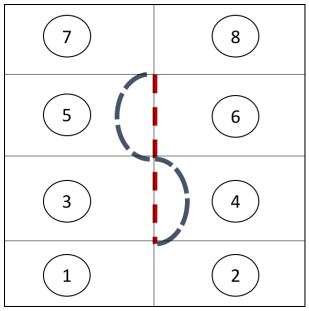


Figure B.1: Patch numbering for studies which incorporate centralized reservoir features. Dashed, colored lines in the center of the domain represent patch boundaries in the case of the simple line sink (red) or the complex line sink (blue).

Table B.1: Multiplicity values for hpk- refinement.

Refinement	Knot Multiplicities
0	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]
1	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]
2	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]
3	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]

Table B.2: NURBS order values for hpk- refinement.

Refinement	NURBS Orders
0	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]
1	[2,2],[2,2],[2,2],[2,2],[2,2],[2,2],[2,2],[2,2]
2	[3,3],[3,3],[3,2],[3,2],[3,2],[3,2],[3,3],[3,3]
3	[4,4],[4,4],[4,2],[4,2],[4,2],[4,2],[4,4],[4,4]

Table B.3: Element number values for hpk- refinement.

Refinement	Number of Elements
0	[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1],[1,1]
1	[2,2],[2,2],[2,1],[2,1],[2,1],[2,1],[2,2],[2,2]
2	[3,3],[3,3],[3,1],[3,1],[3,1],[3,1],[3,3],[3,3]
3	[4,4],[4,4],[4,1],[4,1],[4,1],[4,1],[4,4],[4,4]

B.2 Geometrical Study Reference Solution Parameters

Parameters are presented below for generating the reference solutions to both the simple and complex line sink cases (Table B.4). The patch data may be interpreted similarly to the previous section, although it is important to note that information for the three discussed mesh properties are now condensed into one table since only one refinement level is present.

Table B.4: Patch data for all geometry reference solutions.

Parameters	Data
Multiplicity	[1,1],[1,1],[1,2],[1,2],[1,2],[1,2],[1,1],[1,1]
NURBS order	[8,8],[8,8],[8,2],[8,2],[8,2],[8,2],[8,8],[8,8]
Number Elements	[36,18],[36,18],[36,3],[36,3],[36,3],[36,3],[36,18],[36,18]

Bibliography

Abbas, S., A. Alizada, and T.-P. Fries (2010). The xfem for high-gradient solutions in convection-dominated problems. *International journal for numerical methods in engineering 82*(8), 1044–1072.

Adam, C., T. Hughes, S. Bouabdallah, M. Zarroug, and H. Maitournam (2015). Selective and reduced numerical integrations for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering 284*, 732–761.

Almeida, R. C., R. A. Feijóo, A. C. Galeao, C. Padra, and R. S. Silva (2000). Adaptive finite element computational fluid dynamics using an anisotropic error estimator. *Computer Methods in Applied Mechanics and Engineering 182*(3), 379–400.

Bazilevs, Y., L. Beirao da Veiga, J. A. Cottrell, T. J. Hughes, and G. Sangalli (2006). Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences 16*(07), 1031–1090.

Bazilevs, Y., V. M. Calo, J. A. Cottrell, J. A. Evans, T. Hughes, S. Lipton, M. Scott, and T. Sederberg (2010). Isogeometric analysis using t-splines. *Computer Methods in Applied Mechanics and Engineering 199*(5), 229–263.

Becker, E. B., G. F. Carey, and J. T. Oden (1981). *Finite elements: an introduction*. Englewood Cliffs N.J.: Prentice-Hall.

Berkowitz, B. (2002). Characterizing flow and transport in fractured geological media: A review. *Advances in water resources 25*(8), 861–884.

Borden, M. J., M. A. Scott, J. A. Evans, and T. J. Hughes (2011). Isogeometric finite element data structures based on bézier extraction of nurbs. *International Journal for Numerical Methods in Engineering 87*(1-5), 15–47.

Brooks, A. N. and T. J. Hughes (1982). Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer methods in applied mechanics and engineering 32*(1-3), 199–259.

Brooks, R. H. and A. T. Corey (1964). Hydraulic properties of porous media and their relation to drainage design. *Trans. ASAE 7*(1), 26–0028.

Buckley, S. E. and M. C. Leverett (1942). Mechanism of fluid displacement in sands.

Chen, M.-H., C.-C. Hsu, and W. Shyy (1991). Assessment of tvd schemes for inviscid and turbulent flow computation. *International journal for numerical methods in fluids 12*(2), 161–177.

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

Codina, R. (1998). Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer Methods in Applied Mechanics and Engineering 156*(1-4), 185–210.

Codina, R. (2000). On stabilized finite element methods for linear systems of convection–diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering 188*(1), 61–82.

Cogswell, D. A. and M. L. Szulczewski (2017). Simulation of incompressible two-phase flow in porous media with large timesteps. *Journal of Computational Physics*.

Cottrell, J. A., T. J. Hughes, and Y. Bazilevs (2009). *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons.

De Luycker, E., D. Benson, T. Belytschko, Y. Bazilevs, and M. Hsu (2011). X-fem in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering 87*(6), 541–565.

Finlayson, B. A. (1992). *Numerical methods for problems with moving fronts*. Bruce Alan Finlayson.

Gerke, H. and M. v. Genuchten (1993). A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media. *Water Resources Research 29*(2), 305–319.

Hughes, T. J., J. A. Cottrell, and Y. Bazilevs (2005). Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering 194*(39), 4135–4195.

Hughes, T. J., G. R. Feijóo, L. Mazzei, and J.-B. Quincy (1998). The variational multiscale method—a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering 166*(1-2), 3–24.

Irzal, F., J. Remmers, C. Verhoosel, and R. Borst (2014). An isogeometric analysis bézier interface element for mechanical and poromechanical fracture problems. *International Journal for Numerical Methods in Engineering 97*(8), 608–628.

Jenny, P., H. A. Tchelepi, and S. H. Lee (2009). Unconditionally convergent nonlinear solver for hyperbolic conservation laws with s-shaped flux functions. *Journal of Computational Physics 228*(20), 7497–7512.

Johan, A. H., K. Schrader, et al. (2004). Combination of snake well design & smart completions: Key enablers for champion west development. In *SPE Asia Pacific Oil and Gas Conference and Exhibition*. Society of Petroleum Engineers.

Juanes, R. (2005). A variational multiscale finite element method for multiphase flow in porous media. *Finite elements in analysis and design 41*(7), 763–777.

Juanes, R. and T. W. Patzek (2004). Multiscale-stabilized finite element methods for miscible and immiscible flow in porous media. *Journal of Hydraulic Research 42*(S1), 131–140.

Karimi-Fard, M., L. Durlofsky, K. Aziz, et al. (2003). An efficient discrete fracture model applicable for general purpose reservoir simulators. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

Karimi-Fard, M., A. Firoozabadi, et al. (2001). Numerical simulation of water injection in 2d fractured media using discrete-fracture model. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers.

Katiyar, A., J. T. Foster, H. Ouchi, and M. M. Sharma (2014). A peridynamic formulation of pressure driven convective fluid transport in porous media. *Journal of Computational Physics 261*, 209–229.

Knoll, D. A. and D. E. Keyes (2004). Jacobian-free newton–krylov methods: a survey of approaches and applications. *Journal of Computational Physics 193*(2), 357–397.

Matthäi, S., A. Mezentsev, M. Belayneh, et al. (2005). Control-volume finite-element two-phase flow experiments with fractured rock represented by unstructured 3d hybrid meshes. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers.

McCord, J., M. Reiter, and F. Phillips (1992). Heat-flow data suggest large groundwater fluxes through fruitland coals of the northern san juan basin, colorado-new mexico. *Geology 20*(5), 419–422.

Peters, E. J. (2012). *Advanced Petrophysics: Dispersion, interfacial phenomena*, Volume 2. Greenleaf Book Group.

Piegl, L. and W. Tiller (1997). The nurbs book. *Monographs in Visual Communication*.

Pinder, G. and W. Gray (1977). Finite element simulation in surface and subsurface hydrology.

Sahimi, M. (2011). *Flow and transport in porous media and fractured rock: from classical methods to modern approaches*. John Wiley & Sons.

Sevilla, R., S. Fernández-Méndez, and A. Huerta (2006). Nurbs-enhanced finite element method. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS).

Shahrokhabadi, S., F. Vahedifard, and E. Ghazanfari (2014). Modeling flow regime in shale using isogeometric analysis. In *Shale Energy Engineering 2014@ sTechnical Challenges, Environmental Issues, and Public Policy*, pp. 239–245. ASCE.

Shu, C.-W. (2003). High-order finite difference and finite volume weno schemes and discontinuous galerkin methods for cfd. *International Journal of Computational Fluid Dynamics 17*(2), 107–118.

Tan, M. H., M. Safdari, A. R. Najafi, and P. H. Geubelle (2015). A nurbs-based interface-enriched generalized finite element scheme for the thermal analysis and design of microvascular composites. *Computer Methods in Applied Mechanics and Engineering 283*, 1382–1400.

Thomas, G., D. Thurnau, et al. (1983). Reservoir simulation using an adaptive implicit method. *Society of Petroleum Engineers Journal 23*(05), 759–768.

Warren, J., P. J. Root, et al. (1963). The behavior of naturally fractured reservoirs. *Society of Petroleum Engineers Journal 3*(03), 245–255.

Zimmerman, R. W., G. Chen, T. Hadgu, and G. S. Bodvarsson (1993). A numerical dual-porosity model with semianalytical treatment of fracture/matrix flow. *Water resources research 29*(7), 2127–2137.