**UNIVERSIDADE FEDERAL DE SANTA CATARINA**
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA**
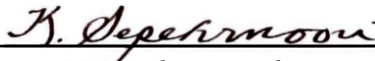
Leonardo Karpinski

**APLICAÇÃO DO MÉTODO DOS VOLUMES FINITOS BASEADO EM ELEMENTOS EM UM SIMULADOR DE RESERVATÓRIOS QUÍMICO-COMPOSICIONAL**

Florianópolis
2011

Leonardo Karpinski

# APLICAÇÃO DO MÉTODO DOS VOLUMES FINITOS BASEADO EM ELEMENTOS EM UM SIMULADOR DE RESERVATÓRIOS QUÍMICO-COMPOSICIONAL

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica para a obtenção do grau de Mestre em Engenharia Mecânica.

Orientador: Clovis Raimundo Maliska
Coorientador: Kamy Sepehrnoori

Florianópolis

2011

Leonardo Karpinski

## APLICAÇÃO DO MÉTODO DOS VOLUMES FINITOS BASEADO EM ELEMENTOS EM UM SIMULADOR DE RESERVATÓRIOS QUÍMICO-COMPOSICIONAL

Esta dissertação foi julgada adequada para a obtenção do título de "Mestre em Engenharia Mecânica", e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Mecânica.

Florianopólis, 11 de Março 2011.

_____
Eduardo Alberto Fancello, D.Sc., Coordenador do Curso

_____
Clovis Raimundo Maliska, Ph.D., Orientador

_____
Kamy Sepehrnoori, Ph.D., Coorientador

**BANCA EXAMINADORA**

_____
António Fábio Carvalho da Silva, Dr.Eng., Presidente

_____
Celso Peres Fernandes, Dr.Eng.

_____
José Antônio Bellini da Cunha Neto, Dr.

# ACKNOWLEDGMENTS

I am sincerely and heartily grateful to my supervisor, Dr. Clovis R. Maliska, for the support and guidance he showed since I joined the SINMEC Lab. I am sure this dissertation would not have been possible without his help. I also would like to thank him for expressing his confidence by choosing me as the first student to develop activities in the cooperation between SINMEC/UFSC and CPGE/UT (Center for Petroleum and Geosystems Engineering at The University of Texas at Austin), where I fulfilled my curricular internship, during my undergraduate studies, and developed part of my Master's research.

I am truly thankful to Dr. Kamy Sepehrnoori, for serving as my co-supervisor from UT Austin and accepting me in this international cooperation. It has been a pleasure to collaborate with a worldwide renowned group, which is the reservoir simulation group at CPGE/UT. I am also grateful to him and his administrative associate for encouraging the use of correct grammar and consistent notation in my writings and for carefully reading and commenting on this manuscript.

My gratitude also goes to Dr. Francisco Marcondes for helping me since the beginning of my research program in Austin. His guidance during the period we were in Austin made my permanence abroad easier.

I deeply appreciate everyone in the SINMEC Lab, for the great work environment. Special thanks to Prof. António Fábio, Fernando Hurtado and Maurício Tada for the long discussions regarding numerical reservoir simulation. Also to Carlos Newmar, for generating the unstructured grids I used in this dissertation.

I would like to show my gratitude to the Post-Graduation Program in Mechanical Engineering (POSMEC) and to the Mechanical Engineering Department of the Federal University of Santa Catarina (UFSC), for giving me all necessary conditions to fulfill this Master's dissertation. I thank the professors from the program, whose expertise and knowledge enabled me to develop a better understanding of the mathematics and physics of fluid

dynamics.

Most importantly, none of this would have been possible without the love and patience of my family, which has been a constant source of love, concern, support and strength all these years, supporting and encouraging my education from the very beginning. I would like to express my gratitude to my parents, Pedro V. Karpinski and Fátima R. P. Karpinski, and to my brother, Pedro Karpinski Neto. Also to Karina G. Fonseca, for being my partner during these two years.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# NOMENCLATURE

| | |
|---|---|
| $b$ | Geometrical and medium information, similar to transmissibility, m$^3$ |
| $\tilde{C}$ | Overall concentration, fraction |
| $C$ | Relative concentration, fraction |
| $\hat{C}$ | Adsorbed concentration, fraction |
| $C^o$ | Component compressibility, Pa$^{-1}$ |
| $C_r$ | Rock compressibility, Pa$^{-1}$ |
| $C_t$ | Total compressibility, Pa$^{-1}$ |
| Cr | Courant number |
| $\vec{\mathbf{D}}$ | Dispersive flux vector, m/s |
| $[D]$ | Matrix of partial derivatives of the shape functions |
| $F$ | Fractional flux |
| $G_w$ | Residue of the water conservation equation, m$^3$/s |
| $h$ | Vertical depth, m |
| $\mathbb{K}$ | Absolute permeability tensor, m$^2$ |
| $k_r$ | Relative permeability |
| $[J]$ | Jacobian matrix |
| $n_{cv}$ | Number of volume-occupying components |
| $n_{ne}$ | Number of nodes per element |
| $n_p$ | Number of phases |
| $\mathcal{N}$ | Element shape functions |
| $P$ | Pressure, Pa |
| $P_c$ | Capillary pressure, Pa |
| $q$ | Volumetric flux, m$^3$/s |
| $\bar{Q}$ | Volumetric injection/production rate per bulk volume, s$^{-1}$ |
| $Q$ | Volumetric injection/production rate, m$^3$/s |

| | |
|---|---|
| $r$ | Reaction rate, $s^{-1}$ |
| $\bar{R}$ | Sink/source term per bulk volume, $s^{-1}$ |
| $S$ | Saturation, fraction |
| $\vec{\mathbf{S}}$ | Surface area vector, $m^2$ |
| $t$ | Time, s |
| $\vec{\mathbf{u}}$ | Darcy's velocity vector, m/s |
| $V$ | Volume, $m^3$ |
| $w$ | Parameter relative to capillary pressure and gravitational terms, Pa |
| $WI$ | Well index, $m^3$ |
| $x, y, z$ | Global coordinates, m |

## Greek symbols

| | |
|---|---|
| $\Delta\vec{\mathbf{S}}$ | Face area vector, $m^2$ |
| $\Delta t$ | Timestep, s |
| $\Delta V$ | Volume of a control volume, $m^3$ |
| $\gamma$ | Specific weight, $N/m^3$ |
| $\lambda$ | Mobility, $(Pa.s)^{-1}$ |
| $\mu$ | Viscosity, Pa.s |
| $\phi$ | Rock porosity, fraction |
| $\rho$ | Density at reservoir conditions over density at standard conditions, fraction |
| $\Theta$ | Generic variable |
| $\varepsilon$ | L2-norm of the numerical solution error |
| $\xi, \eta, \gamma$ | Local coordinates |

## Subscripts

| | |
|---|---|
| 1 | Aqueous phase |
| $e$ | Element |
| $f$ | Face |
| $\dot{f}$ | Integration point, located at the centroid of the face |

| | |
|---|---|
| $\kappa$ | Component |
| $l$ | Phase |
| $p$ | Node |
| ref | Reference condition |
| std | Standard condition |
| $T$ | Total |
| $wf$ | Well |

## Superscripts

| | |
|---|---|
| $n$ | Current time level |
| $n-1$ | Old time level |
| $\theta$ | Generic time level |

## Special symbols

| | |
|---|---|
| ˆ | Nodal value |
| $e \in \mathbb{E}_p$ | Group of elements surrounding node $p$ |
| $f \in \mathbb{F}_p$ | Group of faces in the control surface associated to node $p$ |
| $f \in \mathbb{F}_p^e$ | Group of faces from element $e$ in the control surface associated to node $p$ |
| $k \in \mathbb{T}_p$ | Group of nodes in the stencil of node $p$ |
| $q \in \mathbb{N}^e$ | Group of nodes of element $e$ |

# CONVERSION FACTORS

| | | |
|---|---|---|
| Compressibility | 1 psi$^{-1}$ | $= 1.4504{\times}10^{-4}$ Pa$^{-1}$ |
| Length | 1 ft | $= 0.3048$ m |
| Permeability | 1 mD | $= 9.8692{\times}10^{-16}$ m$^2$ |
| Pressure | 1 psi | $= 6894.7572$ Pa |
| Specific weight | 1 psi/ft | $= 22620.5945$ N/m$^3$ |
| Viscosity | 1 cP | $= 0.001$ Pa.s |
| Volumetric flow rate | 1 ft$^3$/d | $= 3.2774{\times}10^{-7}$ m$^3$/s |

# RESUMO

Este trabalho apresenta uma análise do método dos volumes finitos baseado em elementos (EbFVM) aplicado a um simulador de reservatórios composicional com injeção de componentes químicos, desenvolvido na Universidade do Texas em Austin. O método emprega malhas não estruturadas para a discretização espacial, considerando diferentes tipos de elementos. Elementos dispostos de maneira não estruturada são adequados para representar de forma precisa e eficiente reservatórios com geometrias complexas, combinando a flexibilidade do método dos elementos finitos com a garantia de conservação local e global. Para a obtenção das equações aproximadas para os volumes de controle, primeiramente cada elemento é dividido em sub-volumes de controle, unindo o centróide do elemento com os pontos médios de suas faces. Cada volume de controle é então formado por sub-volumes de controle que compartilham o mesmo vértice, de acordo com o esquema de construção *cell-vertex*. Este procedimento resulta em uma conveniente maneira de gerar malhas capazes de representar importantes aspectos topológicos do reservatório, ao mesmo tempo que preserva-se a essência do método convencional de volumes finitos, ou seja, a construção de equações aproximadas que garantem a conservação de quantidades físicas em nível discreto. A solução das equações foi obtida empregando-se diferentes esquemas de avanço temporal, sendo realizada uma comparação do tempo computacional resultante de cada método, e ilustrando as vantagens e desvantagens de cada um. Para o EbFVM, uma análise do erro das soluções numéricas para diferentes refinos de malha em um problema de difusão com solução analítica conhecida é apresentado, assim como comparações com o método originalmente presente no simulador, onde é mostrado que o EbFVM fornece as mesmas soluções porém com a vantagem de ser um método mais flexível. Por fim, para evidenciar mais claramente as potencialidades do método no quesito geométrico, um problema fisicamente simples em um domínio com geometria não-convencional, possuindo falhas geológicas e poços com trajetórias arbitrárias, é resolvido. Tal geometria não é possível de ser discretizada através do método originalmente presente no simulador, enquanto que com o método proposto isso é facilmente realizado.

# ABSTRACT

An investigation of the Element-based Finite Volume Method (EbFVM) applied to a chemical flooding compositional reservoir simulator, developed at The University of Texas at Austin, is presented. The method employs three-dimensional unstructured grids for the spatial discretization, considering different types of elements. Unstructured elements are adequate to represent, in an accurate and efficient way, reservoirs with complex geometries, combining the flexibility of the finite element method with local and global conservation enforcement. To obtain the approximate equations for the control volumes, first each element is divided into sub-control volumes, by joining the centroid of the element with the midpoint of the faces. Each control volume is then formed by sub-control volumes of neighboring elements applying the cell-vertex construction. This procedure results in a convenient way to build grids that represent important features of the reservoir topology, such as geological faults and wells, at the same time it preserves the essence of conventional finite volume methods, that is, the construction of approximate equations that guarantee the conservation of physical quantities at discrete level. The solution of the equations was obtained by considering different approaches for time advancing. A comparison for the computational time of each approach is performed, illustrating the advantages and disadvantages of each one. For the EbFVM, an analysis of the numerical error for different refinement levels in a diffusion problem with analytical solution is presented, as well as comparisons of the proposed method and the finite difference method originally used in the simulator, by applying different reservoir simulation case studies. It is shown that the EbFVM provides the same solutions but with the advantage of being a more flexible method. Finally, in order to evidence more clearly the element-based potentialities regarding geometry discretization, a physically simple flow problem in a non-conventional geometry, containing geological faults and wells with arbitrary trajectories, is solved. Such geometry is not possible to be discretized by applying the original method present in the simulator, while with the proposed approach it can be done smoothly.

# 1

# **INTRODUCTION**

## 1.1 Preliminaries

Reservoir engineering is one of the important fields in petroleum engineering. In this area, several parameters must be evaluated in order to decide whether a natural reservoir is viable, predicting future performance and finding ways and means of increasing ultimate recovery [3]. Among the techniques used to perform such a study is the application of numerical methods for reservoir simulation, which has gained wide acceptance throughout the petroleum industry in the last thirty years, mostly due to the tremendous increase in computing speed and capacity and to improvements in numerical algorithms for solving partial differential equations [44]. In this respect, reservoir simulation has become the standard for solving reservoir engineering problems [2], and, therefore, has received attention by public and private research centers.

Reservoir simulation is the art of combining physics, mathematics, reservoir engineering, and computer programming to develop a tool for predicting hydrocarbon reservoir performance under various operating strategies. The task of simulating a multiphasic flow problem in porous medium is firstly conceived through a mathematical model formed by a highly coupled system of non-linear partial differential equations. These PDEs, if solved analytically, would provide reservoir pressure, fluid saturations, and well flow rates as continuous functions of space and time [2].

Because of the highly non-linear nature of the PDEs, however, the mathematical model is almost always too complex to be solved by analytical methods. Hence, it is necessary to apply numerical procedures for converting PDEs into a linear system composed by algebraic equations. This process is called discretization, as the solution is obtained for a discrete number of points, where the fundamental equations are applied in their approximate form. Among these numerical procedures, the commonly methods applied are the Finite Difference Method (FDM), Finite Volume Method (FVM), Finite Element Method (FEM) and Element-based Finite Volume Method (EbFVM). It is important to emphasize that the FVM and FDM methods are different [36], however, even so the petroleum literature usually refers to the finite difference method as the discretization approach for the governing equations, while many times the finite volume method is being applied. As this misunderstanding in the designation of the method used to discretize the equations is typical in the literature, it will be kept in the present work intentionally.

The first petroleum simulators employed initially only orthogonal Cartesian grids. The difficulties associated with representing adequately complex reservoir geometries motivated the introduction of non-orthogonal Cartesian grids, known as corner-point grids. This type of grid is nowadays commonly used in commercial simulators, even with the disadvantage of still being structured, which presents difficulties in the geometrical representation as well as in performing local grid refinements in areas of interest, as around wells and faults. Besides, such simulators use inaccurate schemes for the flux calculation when non-orthogonal grids are used [53]. Another discretization option, with more recent applications in petroleum problems, is the use of unstructured grids, similar to the ones used in the finite element method, where a complete geometrical flexibility can be achieved. Figure 1.1 illustrates a reservoir model discretized using the three different approaches cited.

Therefore, the difficulties in the spatial representation of arbitrary domains using structured grids, and the inaccurate techniques of local refinements have motivated the application of unstructured grids in the simulation of reservoir problems. These grids, composed by different element types, are easier to conform to the reservoir domain, increasing the flexibility in representing the main features of the reservoir, such as irregular geometries, wells, and faults. One method that deals with
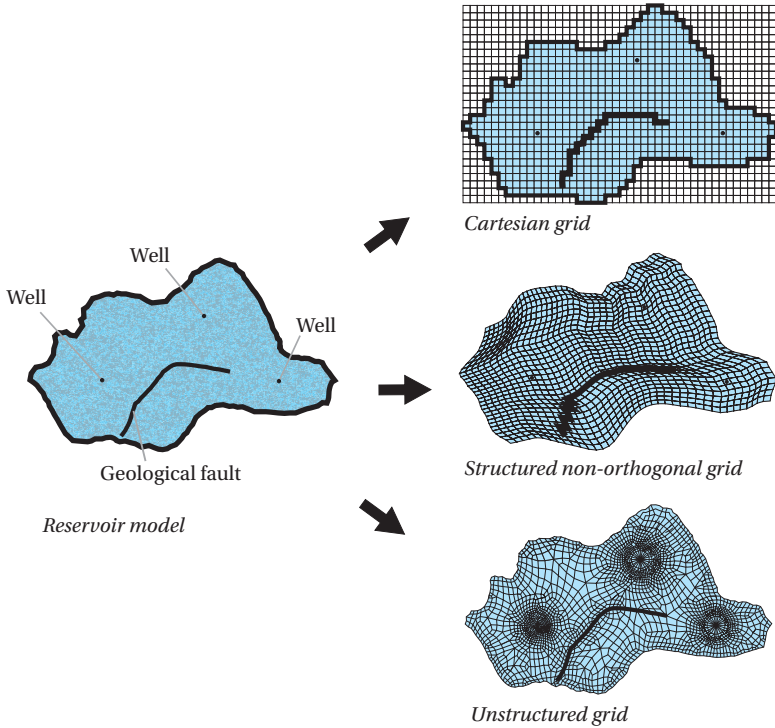
**Figure 1.1** – Reservoir model representation using three different grid types.

unstructured grids is the Element-based Finite Volume Method (EbFVM), which follows the basic idea of the traditional finite volume method, but uses unstructured elements to represent complex geometries and to define the spatial variation of physical media properties. Therefore, it combines the flexibility of the Finite Element Method with local and global conservation enforcement, which is the essence of the FVM. Hence, physical balances are made in the polygonal control volumes, with contributions from different elements, according to the cell vertex construction. This procedure results in a convenient way to build grids with great flexibility, allied to the conservation of physical quantities at discrete level [36].

Although this discretization approach is able to represent all details of geological reservoir models that should be incorporated into the numerical simulations, still little effort has been made in order to take advantage

of numerical formulations that are able to deal with unstructured grids. This dissertation is a sequence of other works performed in the SINMEC Lab [15, 26, 27, 39–42], that are positioned precisely for filling this gap, through the development and implementation of a numerical formulation obtained with the application of a conservative approach based on elements.

## 1.2  Literature review

The numerical technique considered in this work was originally developed for solving the Navier-Stokes equations using unstructured grids in the 80s.  The method arose by the application of the conservation principles directly in a finite element environment, in such a way that the physical properties are conserved in discrete control volumes constructed around the grid nodes. This idea was firstly applied by Baliga and Patankar [5] for triangular elements, while Schneider and Zedan [50] applied it to quadrilateral elements.  These authors denominated the Control Volume Finite Element Method (CVFEM), due to the geometrical flexibility and to the linear system assemblage performed element by element, this denomination being the most known in the literature. However, as argued by Maliska [36], this denomination is inaccurate, as it suggests a finite element formulation that uses control volumes for the integration of the equations. Actually, it is a method conceived according to the same philosophy as the finite volume method, using elements only as geometric supporting entities, by borrowing the concept of element and shape functions from the finite element method. This way, Maliska [36] suggests a more appropriate denomination: Element-based Finite Volume Method (EbFVM), which will be applied in this work.

A few years later, Raw [47] also developed a numerical formulation using quadrilateral elements addressed to fluid dynamics and heat transfer problems.  The form applied in this dissertation derives from the formulation presented in his work, adapting the ideas for solving Navier-Stokes equations to the fluid flow in porous medium.

The first work in the literature using the EbFVM approach for petroleum problems was developed by Rozon [48], which denominated the method as Generalized Finite Volume Method. The author discretized the single-phase equations for quadrilateral elements and compared the

truncation error of this method with other traditional methods. In the beginning of the 90s, other applications of this type of methodology were proposed [22, 24, 25], destined to the simulation of multiphasic flow problems in complex reservoir domains applying unstructured triangular grids. However, some simplifications are introduced in these works, in order to obtain a similar structure as obtained in conventional discretizations with structured grids. As analyzed in Cordazzo *et al.* [16] and Cordazzo *et al.* [17], these simplifications, besides giving rise to an erroneous interpretation of some coefficients in the discretized equations, impose geometrical restrictions to the grid used, that could be avoided if a more rigorous deduction of the approximate equations was considered. More recently, Hurtado [26] presented a formulation for two-phase displacement processes, and Karpinski *et al.* [29] extended such formulation to a chemical flooding compositional reservoir simulator employing two-dimensional hybrid grids.

Once the system of discretized equations is obtained, it is necessary to choose an adequate form for solving it. The simplest method consists in obtaining one equation that the only unknown is the pressure, and algebraic expressions for determining other variables, as saturations, giving rise to the IMPES denomination (IMplicit Pressure Explicit Saturation) in Coats' [11] work. Its basic idea is to obtain a single pressure equation by a combination of the flow equations, updating explicitly the saturations after the pressure has been advanced in time. The major limitation of this approach is the restriction in the timestep size due to the explicit evaluation of saturations. Even with this restriction and with the advent of new techniques for solving linear and non-linear equations, still there is interest in the use of this scheme in certain applications, as well as in the development of strategies for better performance of the method. In this case, the method is improved with the possibility of using larger timesteps that result in solutions free of numerical instabilities [10, 13, 14, 28].

Recently, Chen *et al.* [10] proposed improvements to the IMPES algorithm, with the possibility of taking larger timesteps for the pressure equation than the timesteps used in the saturation equation, as the stability condition is associated only to the explicit approximation of the saturation. This way, the pressure linear system would be solved less times along the simulation, providing a meaningful reduction in the computational time. However, in his work it was not clear how much larger this pressure

timestep can be in relation to the saturation timestep without damaging the numerical solution. Hurtado *et al.* [28] proposed an adaptive control strategy based on the mean variation of the total velocity field as a way of determining a timestep for the pressure equation as large as possible without degrading the quality of the solution. In the demonstrated cases, an expressive time-saving was obtained without introducing further errors in the solutions.

In compositional models, where equations for different components are solved, the IMPSAT model (IMplicit Pressure and SATuration and Explicit Concentration) is usually applied. In this model, only the pressure and saturations are treated implicitly, whereas the concentration of all components are treated explicitly. The method was further analyzed in the last decade, establishing stability criterion and comparing them to the IMPES and Fully Implicit (FIM) models [9]. There is no restriction, however, for using the IMPES approach in compositional simulators. In this case, the correct designation would be IMPEC (IMplicit Pressure and Explicit Concentration), as the pressure and concentrations are the primary variables and the saturation is determined through the phase behavior after solving for pressure implicitly and component concentrations explicitly, as performed in Saad [49].

The fully implicit schemes arrived as an alternative to obtain faster solutions than the IMPES method, especially for finer grids and with regions that present high-speed fluid flow, as its limitations in the timestep size are less restrictive [15]. They consist in the solution of a linear system that involve all variables, usually through the application of Newton's method. Blair and Weinaug [8] were the pioneers in using Newton's method in petroleum reservoir simulation and, since then, this method has been applied by most commercial software.

Another timestepping approach used in reservoir simulation is the sequential method (SEQ), which is situated between the IMPES and FIM methods, with an intermediate level of implicitness. The basic idea of this scheme is to improve the stability of the IMPES method by incorporating implicit treatment of saturations, but without solving simultaneously for pressures and saturations. Therefore, it differs from IMPES only in the manner of the saturation calculations. The pressure distribution, obtained in the same way as in IMPES, is used in a set of flow equations with appropriate linearization of the saturation dependent terms. This way, the

coefficients are treated implicitly in the saturation equation, alleviating the instability of the IMPES method related to the explicit treatment of mobilities, and, therefore, allowing for the use of larger timesteps. Such a scheme was first formulated by MacDonald and Coats [34], but its use was better reported by Spillette *et al.* [52] and Coats [12]. These authors introduced the use of semi-implicit mobility, which proved to be greatly superior to the IMPES and fully implicit method with respect to computational effort, ease of use, and maximum permissible timestep size.

## 1.3 Objectives

As described in the previous sections, the main goal of the present work is the application of the EbFVM formulation to three-dimensional domains in a compositional simulator, called UTCHEM, originally provided by the FDM method. This simulator was developed at The University of Texas at Austin, aiming to simulate enhanced oil recovery processes using chemical flooding. The incorporation of a numerical method able to represent complex geometries with flexibility is an important addition to the simulator, which in terms of mathematical models already accounts for several petroleum engineering options.

Regarding flexibility, hybrid unstructured grids formed by different element types can be easily conformed in order to represent efficiently complex spatial domains. Hexahedral elements allow for precise numerical approximations; however, still there is no method for generating grids composed only by hexahedrons that guarantees satisfactory quality in domains with arbitrary geometry [20, 39]. Hence, hybrid grids composed by hexahedrons and tetrahedrons is a good alternative. The transition between these elements, however, must be performed by elements with similar faces. That is where pyramids and prisms are used, as these elements have both quadrilateral and triangular sides.

Similarly as shown in Maliska *et al.* [39], Figure 1.2 illustrates a possible discretization for a three-dimensional reservoir using a hybrid grid. Some additional details are shown in Figure 1.3, where it is possible to identify different elements in different regions. The cylindrical region around the well is discretized with prisms and hexahedrons, while the faults are represented by hexahedrons, with the rest of the reservoir discretized with
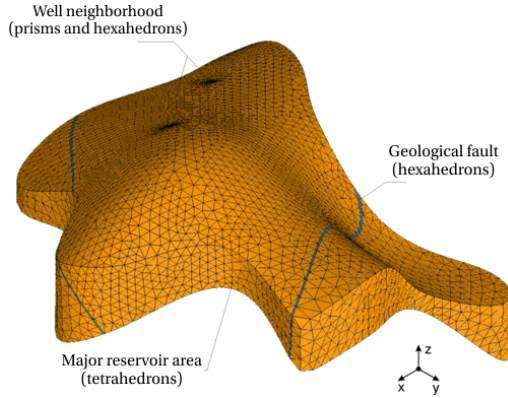
**Figure 1.2** – Hypothetical reservoir discretization using an unstructured grid.



(a) Near-well region

(b) Geological fault
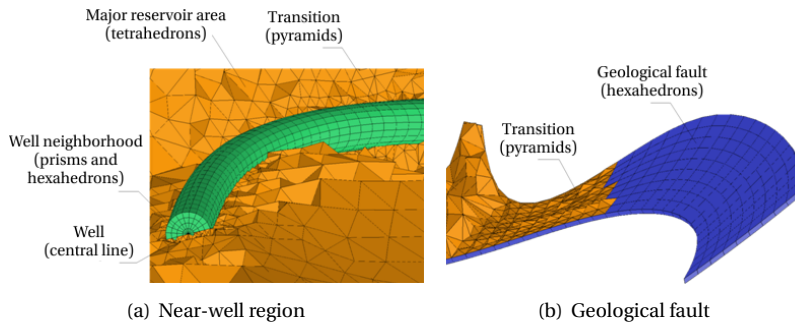
**Figure 1.3** – Discretization of areas of interest.

tetrahedrons, which are easier to be conformed. The transition areas between hexahedrons and tetrahedrons are filled with pyramidal elements.

Besides the application of the EbFVM, different timestepping approaches will be applied in the simulator considered. The original solution method present in UTCHEM is implicit for the pressure and explicit for the concentrations (IMPEC), which, similar to the IMPES method, has a major disadvantage due to the fact that it is highly unstable. Therefore, very small timesteps in order to avoid oscillations in the solutions are necessary. This way, other two approaches will be studied. The first one is still the IMPEC method, but applying different timesteps for the pressure and concentration equations. The second one is the sequential method

(SEQ), which solves implicitly for both the pressure and concentrations, but in a sequential form with two separate linear systems. In this context studies are still necessary for evaluating the performance of such methods, analyzing both the quality of the solutions as well as the computational cost. This activity, jointly with the implementation of the element-based approach in the UTCHEM simulator, compose the central idea of this dissertation, which attempts to realize a comparative evaluation of the FDM and EbFVM methods as well as the time-advancing methods.

## 1.4 Organization of the work

This dissertation is divided into five other chapters. The second chapter describes the UTCHEM simulator, outlining its features and the mathematical model present in the simulator.

Chapter 3 presents several geometrical aspects related to the Element-based Finite Volume Method (EbFVM). In the sequence, the integration of the governing equations is performed, applying such a method.

The different options for treating the time-related variables that emerge during the time integration of the equations are explained in Chapter 4, where the three different timestepping approaches are detailed.

In Chapter 5, the results for different examples are shown. In this chapter, first a validation of the implementation is presented for a diffusion problem. In the sequence, comparisons with the formulation originally existent in the simulator (FDM) are performed. Then the timestepping approaches are applied in conjunction with the EbFVM and FDM methods, comparing the computational cost of each approach. Finally, a simple flow problem in a synthetic reservoir is simulated, in order to demonstrate the potential of the method in discretizing complex domains.

Lastly, Chapter 6 concludes this work with a general discussion regarding the developments made. Some suggestions for future works are also cited.

# 2

# **UTCHEM SIMULATOR**

## 2.1 General description

The simulator considered in this work is called UTCHEM, developed at The University of Texas at Austin to simulate enhanced oil recovery processes and enhanced remediation of aquifers, using chemicals in combination, such as surfactant, polymer, and alkaline flooding [18]. The simulator is a multicomponent, multiphase, and compositional model, accounting for complex phase behavior and multiphase physical properties. Furthermore, it can model capillary pressures, three-phase relative permeabilities (water/gas/oil phases or water/oil/microemulsion phases), dispersion, diffusion, adsorption, chemical reactions, non-equilibrium mass transfer between phases and other related phenomena [55].

In this simulator, the flow and mass-transport equations are solved for any number of user-specified chemical components (water, organic contaminants, surfactant, alcohols, polymer, chloride, calcium, other electrolytes, microbiological species, electron acceptors, etc.). These components can form up to four fluid phases (air, water, oil, and microemulsion) and any number of solid minerals depending on the overall composition. The microemulsion forms only above the critical micelle concentration of the surfactant and is a thermodynamically stable mixture of water, surfactant, and one or more organic components [55]. All of these features taken together, but especially the transport and flow of multiple phases

with multiple species and multiple chemical and biological reactions make UTCHEM unique.

Originally, the solution method is implicit in pressure and explicit in concentration (IMPEC type), and the formulation used to approximate the governing equations is the traditional finite volume method, often called finite difference method in the petroleum literature. As stated before, it is the objective of this work to introduce a method that deals with unstructured grids, the EbFVM in this case, besides the inclusion of different timestepping schemes.

## 2.2 Features

Here some features of the simulator are listed, as in UTCHEM-9.0 [55]:

- Three-dimensional;

- Variable temperature;

- Four phases (water, oil, microemulsion, and gas);

- Heterogeneous permeability and porosity;

- Full tensor dispersion coefficient and molecular diffusion;

- Adsorption of surfactant, polymer, and organic species;

- Solubilization and mobilization of oil;

- Clay/surfactant cation exchange;

- Water/surfactant (cosolvent)/oil phase behavior;

- Polymer with non-Newtonian rheology;

- Compositional density and viscosity functions;

- Surfactant/foam model;

- Multiple organic properties;

- Trapping number including both viscous and buoyancy forces;

- Several polymer/gel kinetics;

- Equilibrium and rate-limited organic dissolution;

- Rock dependent capillary pressure and relative permeability;

- Brooks-Corey capillary pressure and relative permeability functions;

- Water-wet hysteretic capillary pressure and relative permeability model of Parker and Lenhard;

- Mixed-wet hysteretic two-phase oil/water capillary pressure and relative permeability model of Lenhard;

## 2.3 Mathematical model

In this section, a brief description of the mathematical formulation present in the UTCHEM simulator is given. The detailed description of the physical property models and phase behavior can be found elsewhere [18, 55], and additional features needed only for enhanced oil recovery can be found in Bhuyan [7] and Saad [49].

The fundamental equations are the mass balance equation for each species (concentration equation), the aqueous phase pressure, and the energy balance equation (not considered in this work). Therefore, in this work, the primary variables are the component concentrations and the aqueous phase pressure.

The aqueous phase pressure is obtained by an overall mass balance on volume-occupying components (water, oil, surfactant, co-solvent, and gas), while the other phase pressures are computed by adding the capillary pressure between phases. The phases are a single component gas phase ($l = 4$) and up to three liquid phases: aqueous ($l = 1$), oleic ($l = 2$) and microemulsion ($l = 3$), depending on the relative amounts and effective electrolyte concentration (salinity) of the phase environment [55].

### 2.3.1 Concentration equation

The continuity of mass for each component is expressed in terms of overall volume of component $\kappa$ per unit pore volume as

$$\frac{\partial}{\partial t} \left( \phi \, \tilde{C}_\kappa \rho_\kappa \right) + \vec{\nabla} \cdot \left[ \sum_{l=1}^{n_p} \rho_\kappa \left( C_{\kappa l} \, \vec{\mathbf{u}}_l - \vec{\mathbf{D}}_{\kappa l} \right) \right] = \bar{R}_\kappa \; . \tag{2.1}$$

The overall concentration of each component $\tilde{C}_\kappa$ is the sum of the specific concentrations over all phases, including the adsorbed phases:

$$\tilde{C}_\kappa = \left(1 - \sum_{\kappa=1}^{n_{cv}} \hat{C}_\kappa\right) \sum_{l=1}^{n_p} S_l \, C_{\kappa l} + \hat{C}_\kappa \ , \tag{2.2}$$

where $n_{cv}$ is the total number of volume-occupying components, $n_p$ is the number of phases, $S_l$ is the phase saturation, $C_{\kappa l}$ is the concentration of component $\kappa$ in phase $l$, $\hat{C}_\kappa$ is the adsorbed concentration, and $\vec{\mathbf{D}}_{\kappa l}$ is the dispersive flux, assumed to have a Fickian form, as shown in Bear [6] and UTCHEM-9.0 [55]. Also, $\rho_\kappa$ represents a relation between the density of the pure component at reservoir conditions and its density at standard conditions (usually 1 atm), assuming ideal mixing and small and constant component compressibilities $C_\kappa^o$:

$$\rho_\kappa = 1 + C_\kappa^o (P - P_{\text{std}}) \ . \tag{2.3}$$

A similar relation is considered for the rock porosity $\phi$, assuming constant rock compressibility $C_r$:

$$\phi = \phi_{\text{ref}} [1 + C_r (P - P_{\text{ref}})] \ . \tag{2.4}$$

The superficial velocity of each phase is related to the pressure gradient by Darcy's law for multiphase flow, as

$$\vec{\mathbf{u}}_l = -\lambda_l \, \mathbb{K} \cdot \left(\vec{\nabla} P_l - \gamma_l \vec{\nabla} h\right) \ , \tag{2.5}$$

where $P_l$ is the phase pressure, $\mathbb{K}$ is the intrinsic permeability tensor, $h$ is the vertical depth, $\gamma_l$ is the phase specific weight, and $\lambda_l$ is the phase relative mobility, expressed by a relation between the phase relative permeability $k_{rl}$ and the phase viscosity $\mu_l$:

$$\lambda_l = \frac{k_{rl}}{\mu_l} \ . \tag{2.6}$$

The source term $\bar{R}_\kappa$ in Eq. (2.1) is a combination of injection/production volumetric rates and chemical reactions for a particular component, and may be expressed as

$$\bar{R}_\kappa = \phi \sum_{l=1}^{n_p} S_l \, r_{\kappa l} + \left(1 - \phi\right) r_{\kappa s} + \bar{Q}_\kappa \ , \tag{2.7}$$

where $\bar{Q}_\kappa$ is the injection or production rate for component $\kappa$ per bulk volume and $r_{\kappa l}$ and $r_{\kappa s}$ are the reaction rates for component $\kappa$ in phase $l$ and solid phase $s$, respectively.

## 2.3.2 Pressure equation

The pressure equation is obtained by summing the mass balance equations over-all volume-occupying components, substituting Darcy's law for the phase velocity terms and using the definition of capillary pressure. Thus, the pressure equation for the aqueous phase results in

$$\phi_{\text{ref}}\, C_t \frac{\partial P_1}{\partial t} + \vec{\nabla} \cdot \left(\lambda_{Tc}\, \mathbb{K} \cdot \vec{\nabla} P_1\right) =$$
$$\vec{\nabla} \cdot \left[\sum_{l=1}^{n_p} \lambda_{lc}\, \mathbb{K} \cdot \left(\vec{\nabla} P_{cl1} - \gamma_l \vec{\nabla} h\right)\right] + \sum_{\kappa=1}^{n_{cv}} \bar{Q}_\kappa \ , \tag{2.8}$$

where the phase relative mobility including the correction for fluid compressibility $\lambda_{lc}$ is given by

$$\lambda_{lc} = \frac{k_{rl}}{\mu_l} \sum_{\kappa=1}^{n_{cv}} \rho_\kappa C_{\kappa l} \ , \tag{2.9}$$

and the total relative mobility $\lambda_{Tc}$ by

$$\lambda_{Tc} = \sum_{l=1}^{n_p} \lambda_{lc} \ . \tag{2.10}$$

The total compressibility $C_t$ is the volume-weighted sum of the rock ($C_r$) and component compressibilities ($C_\kappa^o$):

$$C_t = C_r + \sum_{\kappa=1}^{n_{cv}} C_\kappa^o \tilde{C}_\kappa \ . \tag{2.11}$$

Note that Eq. (2.8) is not in the conservative form, as the total compressibility term is outside the time derivative. However, as the fluid compressibilities are assumed to be small, there will not be major errors with this action, facilitating the time integration as seen in Chapter 3.

The fundamental equations are, therefore, defined by Eqs. (2.1) and (2.8). As stated before, it is not the objective of this work to detail

how the saturations, viscosities and other physical properties are obtained through the primary variables pressure and concentration. These models are mathematical relations that do not involve geometry, and therefore, do not interfere in the objectives of the present work.

### 2.3.3 Initial and boundary conditions

The initial conditions in UTCHEM are given by an initial pressure and a initial saturation for the aqueous phase, whereas the basic boundary conditions are no convective and no dispersive flux through all boundaries. Alternatively, the no flow condition may be replaced in part by specified pressure on the boundaries, as explained in UTCHEM-9.0 [55]. However, this alternative will not be considered herein.

For the wells, both injection and production wells are considered as source and sink terms in the flow equations. The condition for the wells can be either constant total flow rate or constant bottom hole pressure, and will be further discussed in Chapter 3.

# 3

# **NUMERICAL FORMULATION**

The differential equations that describe the mathematical model presented in the previous chapter have no analytical solution. Thus, in order to solve them numerically, it is necessary to transform these differential equations in algebraic equations, through the application of a numerical method. This transformation process is called discretization, as a continuous problem is transformed into a discrete problem.

The method used to obtain the discrete equations in this work is the Element-based Finite Method (EbFVM), which uses some concepts of the Finite Element Method, but preserves the essential premise of any Finite Volume Method, that is the construction of approximate equations that satisfy the conservation of physical quantities at discrete level.

## 3.1 Main geometrical entities

In the EbFVM formulation, the discretization process is realized considering an unstructured grid. The geometrical entities that define this grid are the *elements*, and they must cover all spots in the domain without superposing other elements or leaving blank spaces. The elements are also used for defining the spatial variation of medium physical properties, as porosity and absolute permeability. This treatment permits to handle heterogeneous permeability distributions in a straightforward way.

The unknowns of the problem are calculated at points called *nodes*, located at every element vertex. Even though the grid is defined by elements, the integration of the governing equations is realized in the *control volumes*, where the conservation of physical quantities is guaranteed. The control volumes are built around grid nodes, with contributions from different elements sharing the same node, so that for each node there is a control volume associated. Therefore, in contrast to traditional finite volume methods, in the EbFVM elements and control volumes no longer coincide. The portion that each element contributes to create a control volume is called *sub-control volume*. In Fig. 3.1 the main geometrical entities of the method are shown for a two-dimensional situation, where the representation is simpler. The same ideas can be easily extended to three dimensions.
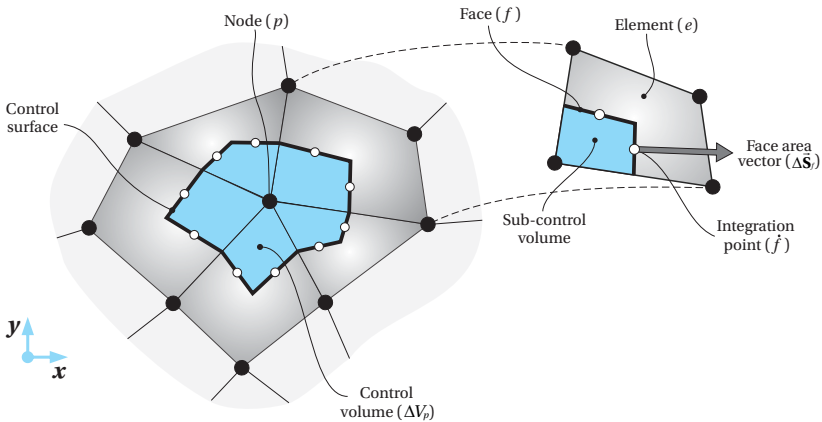


**Figure 3.1** – Main geometrical entities of the EbFVM.

The contour of a control volume, known as *control surface*, is composed by a group of *faces*. In two dimensions, the faces are line segments, as shown in Fig. 3.1, and in three dimensions the faces are triangles or quadrilaterals, as shown in section 3.5. As in any finite volume method, the fluxes that compose the balance of a given physical quantity must be approximated in each face present in the control surface. For this, the best option is to apply the midpoint rule approximation, reason why the midpoint (or centroid) of each face is usually known as *integration point*. Since in this formulation the elements are homogeneous, that

is, properties such as porosity and permeability do not vary inside each element, there is no need to perform any type of averaging procedure for calculating these properties at integration points [15].

## 3.2 Grid definition

Differently from traditional methods using structured grids, where the grid can be defined simply by informing the number of gridblocks[1] and the domain length in each direction, the EbFVM needs a more detailed information of the grid. The easiest way to define it is by informing the coordinates of the nodes and the element connectivity list, which specifies, for each element, the nodes in its vertices, according to the global numeration of the nodes. Figure 3.2 illustrates an example for a 2D grid with four elements and eight nodes.



**Connectivity List**

| Element | Node 1 | Node 2 | Node 3 | Node 4 |
|---------|--------|--------|--------|--------|
| ① | 5 | 6 | 7 | 8 |
| ② | 4 | 5 | 8 | |
| ③ | 2 | 3 | 4 | 8 |
| ④ | 1 | 2 | 8 | 7 |

(c)
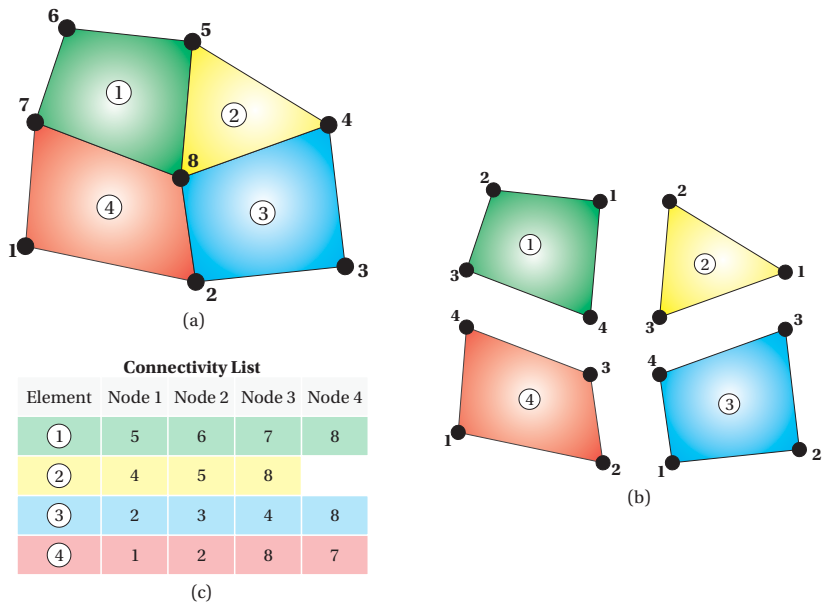
**Figure 3.2** – (a) Global numeration of nodes and elements. (b) Local numeration of the nodes. (c) Connectivity list

---

[1]Gridblocks in the FDM are both the element and the control volume, as these entities coincide in this method. In the EbFVM, the word "gridblock" is not commonly used, as elements and control volumes no longer coincide.

## 3.3 Coordinate transformation

In the EbFVM, the equations can be solved in the computational domain using a standard element in a local coordinate system. This procedure is borrowed from the finite element method, where each element is treated identically and independently, no matter how complex the element geometry is in the global coordinates. In this way, the conservation equations for each control volume can be simply assembled with contributions from the neighboring elements. Figure 3.3 exemplifies the coordinate transformation for a prism.
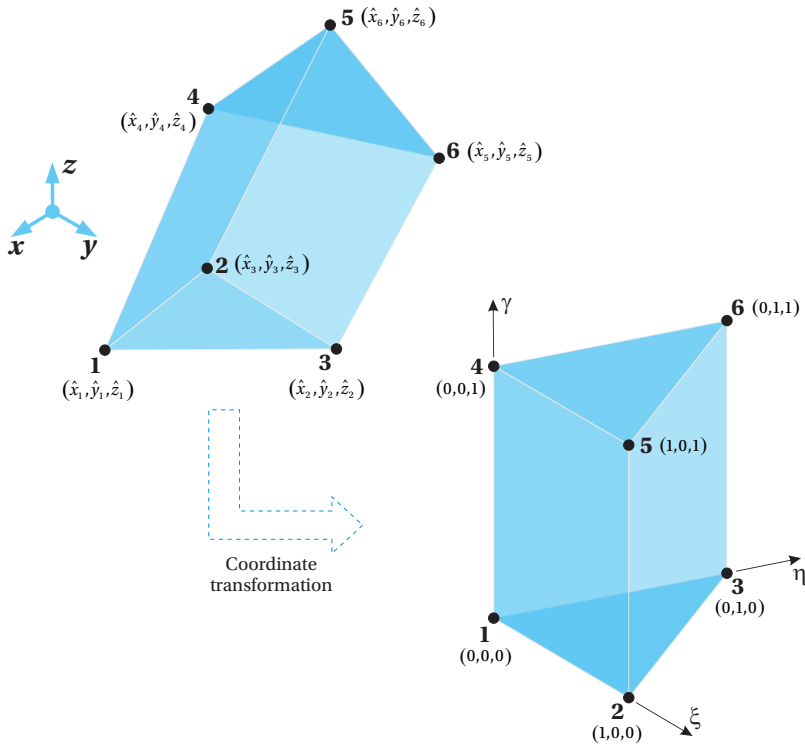


**Figure 3.3** – Element represented in global and local coordinate systems.

Since the conservation equations are defined in respect to the global coordinates $(x, y, z)$, mathematical relations that express the transformation to the local coordinates $(\xi, \eta, \gamma)$ are necessary. This can be done in a

simple way through the use of the element shape functions. Using these functions it is possible to relate the global coordinates of any point inside an element by

$$
\begin{cases}
x\left(\xi,\eta,\gamma\right) = \sum\limits_{k=1}^{n_{ne}} \mathscr{N}_k(\xi,\eta,\gamma)\hat{x}_k \\[2mm]
y\left(\xi,\eta,\gamma\right) = \sum\limits_{k=1}^{n_{ne}} \mathscr{N}_k(\xi,\eta,\gamma)\hat{y}_k \\[2mm]
z\left(\xi,\eta,\gamma\right) = \sum\limits_{k=1}^{n_{ne}} \mathscr{N}_k(\xi,\eta,\gamma)\hat{z}_k \;,
\end{cases}
\tag{3.1}
$$

where $n_{ne}$ is the number of nodes per element, which is variable for each element type, $(\hat{x}_k,\hat{y}_k,\hat{z}_k)$ are the global coordinates of the element nodes and $\mathscr{N}_k(\xi,\eta,\gamma)$ are the shape functions, described for each 3D element in Section 3.5.

## 3.4 Interpolation inside an element

During the discretization process, some physical properties will have to be evaluated inside an element, as for example, the pressure gradient at the integration points for the flux calculation. Knowing the local coordinates of the point of interest, any physical property, denoted here by $\Theta$, can be expressed in terms of the element shape functions and the property values in the element vertices. That is,

$$
\Theta\left(\xi,\eta,\gamma\right) = \sum_{k=1}^{n_{ne}} \mathscr{N}_k(\xi,\eta,\gamma)\hat{\Theta}_k \;,
\tag{3.2}
$$

where $\hat{\Theta}_k$ are the nodal values for the property. In a more compact way, Eq. (3.2) can be written as

$$
\Theta = [\mathscr{N}]^T [\hat{\Theta}]_e \;,
\tag{3.3}
$$

where $[\hat{\Theta}]_e = [\ \hat{\Theta}_1 \quad \hat{\Theta}_2 \quad \cdots \quad \hat{\Theta}_{n_{ne}}\ ]^T$ is the vector containing the nodal values of the property and $[\mathscr{N}]^T = [\ \mathscr{N}_1 \quad \mathscr{N}_2 \quad \cdots \quad \mathscr{N}_{n_{ne}}\ ]$ is the vector containing the element shape functions.

Based on the approximation given in Eq. (3.2), it is possible to express the gradient of the variable of interest in terms of the local coordinate

system. Differentiating this equation in relation to the global coordinates, the gradient vector can be written as

$$[\nabla\Theta] \equiv \begin{bmatrix} \partial_x \Theta \\ \partial_y \Theta \\ \partial_z \Theta \end{bmatrix} = \sum_{k=1}^{n_{ne}} \begin{bmatrix} \partial_x \mathcal{N}_k \\ \partial_y \mathcal{N}_k \\ \partial_z \mathcal{N}_k \end{bmatrix} \hat{\Theta}_k \ , \tag{3.4}$$

or in another form as

$$[\nabla\Theta] = \begin{bmatrix} \partial_x \mathcal{N}_1 & \partial_x \mathcal{N}_2 & \cdots & \partial_x \mathcal{N}_{n_{ne}} \\ \partial_y \mathcal{N}_1 & \partial_y \mathcal{N}_2 & \cdots & \partial_y \mathcal{N}_{n_{ne}} \\ \partial_z \mathcal{N}_1 & \partial_z \mathcal{N}_2 & \cdots & \partial_z \mathcal{N}_{n_{ne}} \end{bmatrix} \begin{bmatrix} \hat{\Theta}_1 \\ \hat{\Theta}_2 \\ \vdots \\ \hat{\Theta}_{n_{ne}} \end{bmatrix} . \tag{3.5}$$

The key point is the determination of the shape function derivatives in relation to the global coordinates in Eq. (3.5). As the shape functions are continuous functions inside an element, they can be differentiated. Considering the chain rule, it is possible to write

$$\begin{bmatrix} \partial_\xi \mathcal{N}_k \\ \partial_\eta \mathcal{N}_k \\ \partial_\gamma \mathcal{N}_k \end{bmatrix} = \begin{bmatrix} \partial_\xi x & \partial_\xi y & \partial_\xi z \\ \partial_\eta x & \partial_\eta y & \partial_\eta z \\ \partial_\gamma x & \partial_\gamma y & \partial_\gamma z \end{bmatrix} \begin{bmatrix} \partial_x \mathcal{N}_k \\ \partial_y \mathcal{N}_k \\ \partial_z \mathcal{N}_k \end{bmatrix} , \tag{3.6}$$

where the matrix with dimensions $3 \times 3$ in the right-hand side is known as the Jacobian of the coordinate transformation, usually denoted as $[J]$. This equation can be rewritten, then, in a more compact form as

$$\begin{bmatrix} \partial_\xi \mathcal{N}_k \\ \partial_\eta \mathcal{N}_k \\ \partial_\gamma \mathcal{N}_k \end{bmatrix} = [J] \begin{bmatrix} \partial_x \mathcal{N}_k \\ \partial_y \mathcal{N}_k \\ \partial_z \mathcal{N}_k \end{bmatrix} . \tag{3.7}$$

The Jacobian matrix can be easily calculated by

$$[J] = [D] [\Omega]_e \ , \tag{3.8}$$

where $[D]$ is the first-order partial derivatives of the shape functions written in a matrix form as

$$[D] = \begin{bmatrix} \partial_\xi \mathscr{N}_1 & \partial_\xi \mathscr{N}_2 & \cdots & \partial_\xi \mathscr{N}_{n_{ne}} \\ \partial_\eta \mathscr{N}_1 & \partial_\eta \mathscr{N}_2 & \cdots & \partial_\eta \mathscr{N}_{n_{ne}} \\ \partial_\gamma \mathscr{N}_1 & \partial_\gamma \mathscr{N}_2 & \cdots & \partial_\gamma \mathscr{N}_{n_{ne}} \end{bmatrix} , \tag{3.9}$$

and $[\Omega]_e$ is a matrix containing the global coordinates of the nodes in the vertices of element $e$, as the form

$$[\Omega]_e = \begin{bmatrix} \hat{x}_1 & \hat{y}_1 & \hat{z}_1 \\ \hat{x}_2 & \hat{y}_2 & \hat{z}_2 \\ \vdots & \vdots & \vdots \\ \hat{x}_{n_{ne}} & \hat{y}_{n_{ne}} & \hat{z}_{n_{ne}} \end{bmatrix} . \tag{3.10}$$

After determining the Jacobian matrix with Eq. (3.8), the derivatives of the shape functions in relation to the global coordinates can be calculated by multiplying both sides of Eq. (3.7) by the inverse of $[J]$,

$$[J]^{-1} \begin{bmatrix} \partial_\xi \mathscr{N}_k \\ \partial_\eta \mathscr{N}_k \\ \partial_\gamma \mathscr{N}_k \end{bmatrix} = \begin{bmatrix} \partial_x \mathscr{N}_k \\ \partial_y \mathscr{N}_k \\ \partial_z \mathscr{N}_k \end{bmatrix} . \tag{3.11}$$

Extending this equation to all $n_{ne}$ nodes and commuting the sides, one obtains

$$\begin{bmatrix} \partial_x \mathscr{N}_1 & \partial_x \mathscr{N}_2 & \cdots & \partial_x \mathscr{N}_{n_{ne}} \\ \partial_y \mathscr{N}_1 & \partial_y \mathscr{N}_2 & \cdots & \partial_y \mathscr{N}_{n_{ne}} \\ \partial_z \mathscr{N}_1 & \partial_z \mathscr{N}_2 & \cdots & \partial_z \mathscr{N}_{n_{ne}} \end{bmatrix} = [J]^{-1} [D] . \tag{3.12}$$

Finally, substituting into Eq. (3.5), the gradient of the variable of interest is given by

$$[\nabla \Theta] = [J]^{-1} [D] [\hat{\Theta}]_e . \tag{3.13}$$

Equation (3.13) approximates the gradient of the variable of interest at any point $(\xi, \eta, \gamma)$ inside an element. This equation will be used to approximate the pressure gradient during the integration process in section 3.6.

## 3.5 Three-dimensional elements

In this section, the shape functions of the four different three-dimensional elements considered is presented. Other geometrical characteristics, for instance, the calculation of the area of the faces and the volume of the sub-control volumes can be found elsewhere [27, 40]. It is important to note that the number of sub-control volumes of each element is equal to the number of vertices, and the number of faces[2] is equal to the number of edges. The elements considered in this work are the following:

- Tetrahedron, with four vertices, six edges, and four triangular sides.

- Hexahedron, with eight vertices, twelve edges, and six quadrilateral sides.

- Prism, with six vertices, nine edges and five sides, where two are triangular and three are quadrilateral.

- Pyramid, with five vertices, eight edges and five sides, where four are triangular and one is quadrilateral.

As there is still no method for generating arbitrary unstructured grids only with hexahedrons, hybrid grids with hexahedrons and tetrahedrons are a good alternative. However, prisms and pyramids are necessary to perform the transition between tetrahedrons and hexahedrons, in order to avoid non-conformity of the grid, as two neighboring elements must have one side with the same form, triangular or quadrilateral.

In three dimensions, the construction of the control volume is equivalent to the two-dimensional situation. Each element is first divided into a certain number of sub-control volumes, according to its number of vertices, and then the control volume is assembled with the sub-control volumes surrounding the same node. As in two-dimensions, there is one control volume associated to each grid node. The main ideas employed here were presented in Maliska *et al.* [40].

---

[2]According to the nomenclature adopted in this work, the *faces* always refer to the interfaces where the fluxes are calculated and not to the faces in the element's external sides.

### 3.5.1 Tetrahedron

The shape functions of the tetrahedron are linear functions; thus, the Jacobian matrix is constant along the element. Table 3.1 shows the definition of the four shape functions for this type of element.

**Table 3.1** – Tetrahedron shape functions.

| $\mathcal{N}_k(\xi,\eta,\gamma)$ | |
|:---:|:---:|
| $\mathcal{N}_1$ | $1-\xi-\eta-\gamma$ |
| $\mathcal{N}_2$ | $\xi$ |
| $\mathcal{N}_3$ | $\eta$ |
| $\mathcal{N}_4$ | $\gamma$ |

In the sequence, Fig. 3.4 shows the standard tetrahedron and the local coordinates of its vertices.



| Node | $\xi$ | $\eta$ | $\gamma$ |
|:---:|:---:|:---:|:---:|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 |
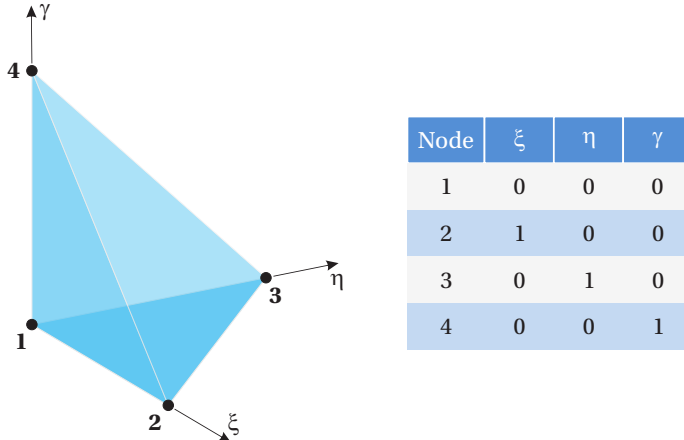
**Figure 3.4** – Standard tetrahedron and local coordinates of the nodes.

The tetrahedron division generates four sub-control volumes. All of them are irregular hexahedrons, limited by six quadrilateral sides, where three of them are inner faces. Figure 3.5 illustrates these faces for one sub-control volume of the element. The centroid coordinate of the faces for

this and other elements, as well as other aspects relative to sub-control volumes and faces can be found in Maliska *et al.* [40] and Hurtado [27].
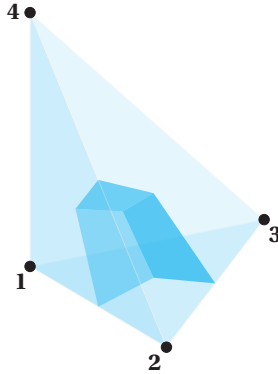


**Figure 3.5** – Faces in standard tetrahedron.

## 3.5.2 Hexahedron

Table 3.2 shows the definition of the shape functions for the hexahedron element. They are tri-linear functions of the local coordinates.

**Table 3.2** – Hexahedron shape functions.

| $\mathcal{N}_k(\xi, \eta, \gamma)$ | |
|---|---|
| $\mathcal{N}_1$ | $(1-\xi)(1-\eta)(1-\gamma)$ |
| $\mathcal{N}_2$ | $\xi(1-\eta)(1-\gamma)$ |
| $\mathcal{N}_3$ | $\xi\eta(1-\gamma)$ |
| $\mathcal{N}_4$ | $(1-\xi)\eta(1-\gamma)$ |
| $\mathcal{N}_5$ | $(1-\xi)(1-\eta)\gamma$ |
| $\mathcal{N}_6$ | $\xi(1-\eta)\gamma$ |
| $\mathcal{N}_7$ | $\xi\eta\gamma$ |
| $\mathcal{N}_8$ | $(1-\xi)\eta\gamma$ |

The standard hexahedron is a regular cube, with eight cubic sub-control volumes. Figure 3.6 illustrates the standard hexahedron and the local coordinates of the nodes.



| Node | $\xi$ | $\eta$ | $\gamma$ |
|------|-------|--------|----------|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | 0 | 1 | 0 |
| 5 | 0 | 0 | 1 |
| 6 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 |
| 8 | 0 | 1 | 1 |
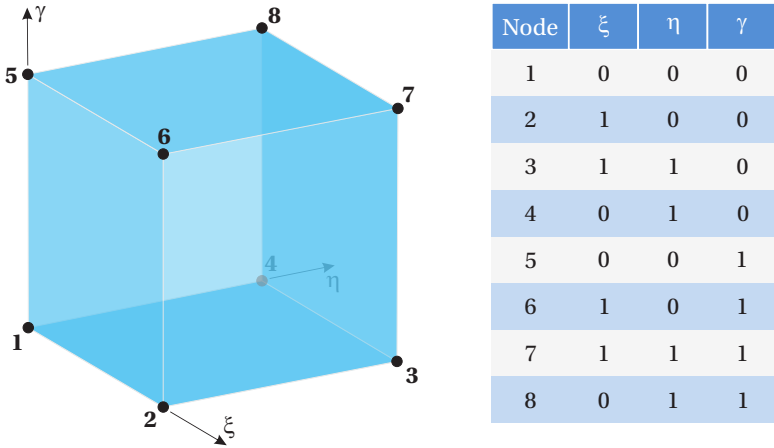
**Figure 3.6** – Standard hexahedron and local coordinates of the nodes.

All twelve inner faces are quadrilateral, mutually perpendicular. Figure 3.7 shows the three faces for the sub-control volume associated to node 2.
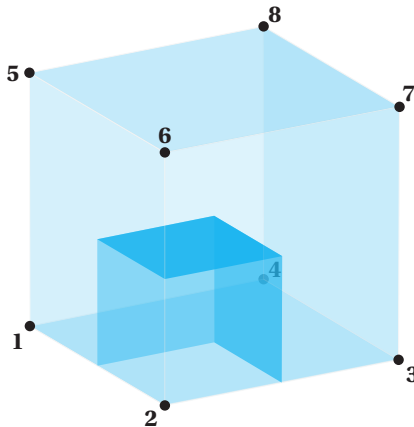


**Figure 3.7** – Faces in standard hexahedron.

### 3.5.3 Prism

The prism is used as a transition element between tetrahedrons and hexahedrons, as it has both triangular and quadrilateral sides. Its shape functions are defined in Table 3.3.

**Table 3.3** – Prism shape functions.

| $\mathcal{N}_k(\xi,\eta,\gamma)$ | |
|---|---|
| $\mathcal{N}_1$ | $(1-\xi-\eta)(1-\gamma)$ |
| $\mathcal{N}_2$ | $\xi(1-\gamma)$ |
| $\mathcal{N}_3$ | $\eta(1-\gamma)$ |
| $\mathcal{N}_4$ | $(1-\xi-\eta)\gamma$ |
| $\mathcal{N}_5$ | $\xi\gamma$ |
| $\mathcal{N}_6$ | $\eta\gamma$ |

The standard prism and the coordinates of the nodes in the transformed space are shown in Fig. 3.8.



| Node | $\xi$ | $\eta$ | $\gamma$ |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 |
| 6 | 0 | 1 | 1 |

**Figure 3.8** – Standard prism and local coordinates of the nodes.

As in the tetrahedron, the sub-control volumes in the prism are irregular hexahedrons. Figure 3.9 illustrates the faces for one sub-control

volume.
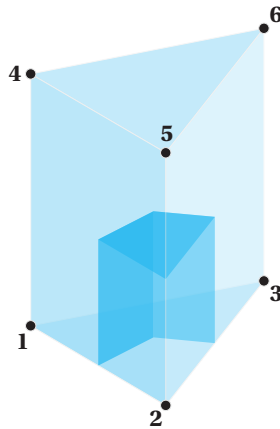


**Figure 3.9** – Faces in standard prism.

## 3.5.4 Pyramid

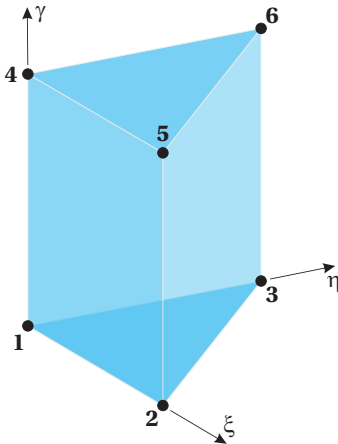As well as the prism, the pyramid is a transition element between tetrahedrons and hexahedrons, as it has four triangular sides and a quadrilateral base. It needs, however, non-polynomial shape functions, in order to guarantee the continuity of the variable with the other elements [40]. The shape functions are listed in Table 3.4, as in Maliska *et al.* [40] and Zgainski *et al.* [58].

**Table 3.4** – Pyramid shape functions.

| $\mathcal{N}_k(\xi, \eta, \gamma)$ | |
|:---:|:---:|
| $\mathcal{N}_1$ | $\frac{1}{4}\left[(1-\xi)(1-\eta)-\gamma+\xi\eta\gamma/(1-\gamma)\right]$ |
| $\mathcal{N}_2$ | $\frac{1}{4}\left[(1+\xi)(1-\eta)-\gamma+\xi\eta\gamma/(1-\gamma)\right]$ |
| $\mathcal{N}_3$ | $\frac{1}{4}\left[(1+\xi)(1+\eta)-\gamma+\xi\eta\gamma/(1-\gamma)\right]$ |
| $\mathcal{N}_4$ | $\frac{1}{4}\left[(1-\xi)(1+\eta)-\gamma+\xi\eta\gamma/(1-\gamma)\right]$ |
| $\mathcal{N}_5$ | $\gamma$ |

In Fig. 3.10 the standard pyramid and the local coordinates of its vertices are shown. Differently from the other elements, coordinates $\xi$ and $\eta$ can be negative.



| Node | $\xi$ | $\eta$ | $\gamma$ |
|------|-------|--------|----------|
| 1 | -1 | -1 | 0 |
| 2 | 1 | -1 | 0 |
| 3 | 1 | 1 | 0 |
| 4 | -1 | 0 | 1 |
| 5 | 0 | 0 | 1 |
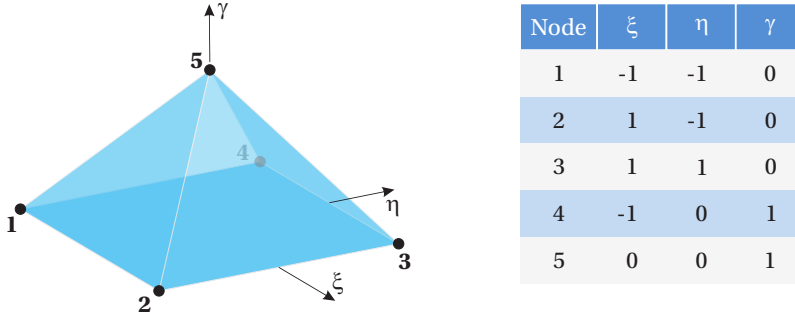
**Figure 3.10** – Standard pyramid and local coordinates of the nodes.

The way to build the pyramid sub-control volumes differs from that of the previous elements. In the pyramid, all sub-control volumes have the base centroid as a common point, while for the other elements the common point is the element centroid. As a consequence, the inner faces connecting the nodes in the base of the pyramid are triangular, while the faces connecting the apex with the base are quadrilateral, as noted in Fig. 3.11 in the division for one sub-control volume.
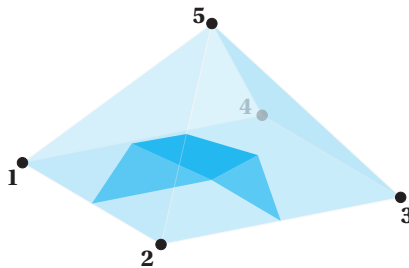


**Figure 3.11** – Faces in standard pyramid.

## 3.6  Integration of the governing equations

This section describes the application of the EbFVM in the discretization process. In order to obtain the mass conservation for each control

volume, it is necessary to integrate Eqs. (2.1) and (2.8) in space and time, considering a control volume as the integration domain.

### 3.6.1 Pressure equation

First, the integration of the pressure equation leads to

$$
\iint_{\Delta t \, \Delta V} \phi_{\text{ref}} \, C_t \, \frac{\partial P_1}{\partial t} \, dV \, dt + \iint_{\Delta t \, \Delta V} \vec{\nabla} \cdot \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right) dV \, dt =
$$
$$
\iint_{\Delta t \, \Delta V} \vec{\nabla} \cdot \left[ \sum_{l=1}^{n_p} \lambda_{lc} \, \mathbb{K} \cdot \left( \vec{\nabla} P_{cl1} - \gamma_l \vec{\nabla} h \right) \right] dV \, dt + \iint_{\Delta t \, \Delta V} \sum_{\kappa=1}^{n_{cv}} \bar{Q}_\kappa \, dV \, dt \ .
$$

(3.14)

Applying the divergence theorem to Eq. (3.14) for both the second term of the LHS and the first term of the RHS, we obtain

$$
\iint_{\Delta t \, \Delta V} \phi_{\text{ref}} \, C_t \, \frac{\partial P_1}{\partial t} \, dV \, dt + \iint_{\Delta t \, \Delta S} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right) \cdot \mathbf{d\vec{S}} \, dt =
$$
$$
\iint_{\Delta t \, \Delta S} \sum_{l=1}^{n_p} \left( \lambda_{lc} \, \mathbb{K} \cdot \vec{\nabla} w_l \right) \cdot \mathbf{d\vec{S}} \, dt + \iint_{\Delta t \, \Delta V} \sum_{\kappa=1}^{n_{cv}} \bar{Q}_\kappa \, dV \, dt \ ,
$$

(3.15)

where $\vec{\nabla} w_l = \left( \vec{\nabla} P_{cl1} - \gamma_l \vec{\nabla} h \right)$ and $\mathbf{d\vec{S}}$ is the differential area vector, normal to the control volume contour in any point and pointing outside the volume. The surface integrals are performed over the control surface.

Considering a polyhedric control volume, the surface integrals in Eq. (3.15) can be expressed as the sum of the integrals over all faces in the control surface. That is,

$$
\int_{\Delta S} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right) \cdot \mathbf{d\vec{S}} = \sum_{f \in \mathbb{F}_p} \int_{\Delta S_f} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right) \cdot \mathbf{d\vec{S}} \ ,
$$

(3.16)

where $f \in \mathbb{F}_p$ are all faces in the control surface, following the notation used in Maliska *et al.* [40]. Approximating these integrals by means of the midpoint rule [3]:

$$\sum_{f \in \mathbb{F}_p} \int_{\Delta S_f} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right) \cdot \mathbf{d\vec{S}} = \sum_{f \in \mathbb{F}_p} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right)_{\hat{f}} \cdot \Delta \vec{\mathbf{S}}_f \, , \qquad (3.17)$$

where $\hat{f}$ represents the integration point located at the centroid of face $f$ and $\Delta \vec{\mathbf{S}}_f$ denotes the face area vector, pointing outside of the control volume.

Substituting Eq. (3.17) into Eq. (3.15) and applying the same procedure for the other term containing a surface integral, we obtain

$$\int_{\Delta t} \left( \phi_{\text{ref}} \, \Delta V \, C_t \frac{\partial P_1}{\partial t} \right)_p dt + \int_{\Delta t} \left[ \sum_{f \in \mathbb{F}_p} \left( \lambda_{Tc} \, \mathbb{K} \cdot \vec{\nabla} P_1 \right)_{\hat{f}} \cdot \Delta \vec{\mathbf{S}}_f \right] dt =$$

$$\int_{\Delta t} \sum_{f \in \mathbb{F}_p} \left[ \left( \sum_{l=1}^{n_p} \lambda_{lc} \, \mathbb{K} \cdot \vec{\nabla} w_l \right)_{\hat{f}} \cdot \Delta \vec{\mathbf{S}}_f \right] dt + \int_{\Delta t} \left( \sum_{\kappa=1}^{n_{cv}} Q_\kappa \right)_p dt \, , \tag{3.18}$$

where $Q_\kappa = \bar{Q}_\kappa \, \Delta V$.

Equation (3.18) refers to the control volume around node $p$, with volume equal to $\Delta V_p$, and is formed by sub-control volumes from different elements. In three-dimensional domains each sub-control volume contributes with three faces $f$ over a control surface. The exception is the sub-control associated to the apex of a pyramid, which contributes to the control volume with four faces.

For the time integration, all terms with the exception of the pressure time derivative will be considered constant during the time $\Delta t$. Therefore, they have to be evaluated between the old time level $(n-1)$ and the current time level $(n)$, where the mobilities are unknowns. The most complex case is when all time levels are chosen equal to the current time level $n$, generating a fully implicit scheme. This will not be considered in the present work, as discussed in Chapter 4. In this work, the pressure and the concentration equations are decoupled. Therefore, the terms that are dependent on the concentration will be evaluated at time level $n-1$. Note that the total compressibility depends on the component concentrations, and, therefore, is known only at the old time level. Thus, Eq. (3.18) can be written in a matrix form as

$$\left[ \phi_{\text{ref}} \Delta V \, C_t^{n-1} \frac{\left( P_1^n - P_1^{n-1} \right)}{\Delta t} \right]_p +$$

$$\sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} (\lambda_{Tc})_f^{n-1} [\Delta S]_f^T [K]_e [\nabla P_1]_f^n =$$

$$\sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} \sum_{l=1}^{n_p} (\lambda_{lc})_f^{n-1} [\Delta S]_f^T [K]_e [\nabla w_l]_f^{n-1} + \sum_{\kappa=1}^{n_{cv}} \left( Q_\kappa^{n^*} \right)_p ,$$

(3.19)

where $e \in \mathbb{E}_p$ are all elements that are around node $p$, contributing to the control volume formation, $f \in \mathbb{F}_p^e$ are the faces from element $e$ in the control surface and $\Delta t$ is the timestep. Note that in Eq. (3.18) the summation is over $f \in \mathbb{F}_p$ and here it was split in two summations, as in Maliska *et al.* [40]. Another point is that the subscript of the well flow term $Q_\kappa$, defined posteriorly in Eq. (3.25), is $n^*$. Actually, when the well is producing at a specified bottom hole pressure the phase mobility term is evaluated at time level $n$-1 and the pressure difference term is evaluated at time level $n$.

It is necessary to introduce an approximation for the gradients present in Eq. (3.19). The conventional approach in the EbFVM is to assume local variation in each element according to the same functions used in the coordinate transformation. Therefore, Eq. (3.13) can be used directly here, and Eq. (3.19) can be written as

$$\left[ \phi_{\text{ref}} \Delta V \, C_t^{n-1} \frac{\left( P_1^n - P_1^{n-1} \right)}{\Delta t} \right]_p + \sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} (\lambda_{Tc})_f^{n-1} [b]_f^T \left[ \hat{P}_1 \right]_e^n =$$

$$\sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} \sum_{l=1}^{n_p} (\lambda_{lc})_f^{n-1} [b]_f^T [\hat{w}_l]_e^{n-1} + \sum_{\kappa=1}^{n_{cv}} \left( Q_\kappa^{n^*} \right)_p ,$$

(3.20)

where $[\hat{w}_l]_e$ is defined as

$$[\hat{w}_l]_e = \left[ \hat{P}_{cl1} \right]_e - \left[ \hat{\gamma}_l h \right]_e ,$$

(3.21)

and $[b]_f^T$ is a vector defined for each face $f$ inside element $e$ as

$$[b]_{\hat{f}}^{T} = [\Delta S]_{\hat{f}}^{T} [K]_e [J]_{\hat{f}}^{-1} [D]_{\hat{f}} \;. \tag{3.22}$$

This operator has the similar function of the transmissibilities in conventional finite volume methods, that is, a coefficient to connect the pressure between neighboring control volumes. As it is a function only of the geometry and medium, and these entities do not vary along the simulation, it is calculated only once in the beginning of the simulation.

### 3.6.2 Concentration equation

Applying the same approach for the concentration equation, the following discrete equation is obtained:

$$\left[ \phi_{\text{ref}} \, \Delta V \frac{\left( \tilde{C}_{\kappa}^{n} - \tilde{C}_{\kappa}^{n-1} \right)}{\Delta t} \right]_p =$$
$$\sum_{f \in \mathbb{F}_p} \sum_{l=1}^{n_p} \left[ (C_{\kappa l})_{\hat{f}}^{\theta} \, (q_l)_{\hat{f}}^{\theta} + (D_{\kappa l})_{\hat{f}}^{\theta} \right] + \left( \frac{R_{\kappa}^{\theta}}{\rho_{\kappa}} \right)_p , \tag{3.23}$$

where $(q_l)_{\hat{f}}$ is the flux of phase $l$ across face $f$, defined as

$$(q_l)_{\hat{f}}^{\theta} = (\lambda_{lc})_{\hat{f}}^{\theta} \, [b]_{\hat{f}}^{T} \left\{ \left[ \hat{P}_l \right]_e^{n} - \left[ \gamma_l \hat{h} \right]_e^{\theta} \right\} \;. \tag{3.24}$$

Here, the time level where properties are evaluated is represented by $\theta$ for all terms except the phase pressure term. As stated before, in this work the pressure and the concentration are decoupled, so that when the concentration is solved the pressure at time level $n$ is already determined. This discussion is further detailed in the next chapter.

## 3.7 Well models

The realistic representation of wells in numerical formulations has always been a challenging problem in reservoir simulation. The main problem resides in the difference in scale between the well, order of centimeters, and the reservoir's order of kilometers. With these scales, in order to model the well and to capture the real pressure gradient in its vicinity, an extremely fine grid would be necessary. However, this kind of approach is

not usual, since it would demand high computational effort. Therefore, an analytical mathematical model for calculating a local solution is the usual approach to couple the well variables, as pressure and flow rate, with the reservoir variables. This approach is known as well model, which provides the well index coefficient, and when formulated with too many restrictions results in poor and non-realistic approximations to the local problem [54].

As stated previously, both injection and production wells are considered source and sink terms in the flow equations. The condition for the wells can be either constant volumetric flow rate or constant bottom hole pressure. In the first case, the flow rate $Q_\kappa$ present in the integrated equations is equal to the value prescribed. For the second case, the flow rate must be related to the pressure gradient between well and reservoir [46], and may be expressed as

$$Q_\kappa = \sum_{l=1}^{n_p} C_{\kappa l} \lambda_l \, WI \left( P_{wf} - P_l \right) \ .$$

(3.25)

In this equation, $WI$ is the well index coefficient, which is basically a function of geometry and rock properties, and relates the well pressure ($P_{wf}$) with the wellblock[3] pressure ($P_l$). This parameter plays a similar role as the operator $[G]_f^T$ defined in Eq. (3.22), even with the same unity, $[L^3]$.

The well models considered in UTCHEM are based on formulations by Peaceman [46] and Babu and Odeh [4]. Even though these models are based on several assumptions, they are often used in situations where the assumptions are not respected, as more accurate models are not easy to obtain. Recently, several authors [1, 19, 35, 57] developed methodologies that calculate the well index for arbitrary well configurations so that the classical model assumptions can be avoided. These models are more complex than the common ones, as they are based on more complete analytical solutions in the near well region, providing a less restrictive model. The formulation proposed by Maizeret [35], Ouyang and Aziz [43] and Wolfsteiner *et al.* [57], that uses Green's function for obtaining the analytical solution, is explained with details in Karpinski *et al.* [30] and Maliska *et al.* [38].

---

[3]In traditional methods with structured grids, the well is located in the center of the grid-blocks, and a gridblock containing a well is the denominated wellblock. In this dissertation, the well is located in the grid nodes, and the wellblock in this case will represent the control volume containing a well in its node.

It is not the objective of this work to implement and use a complex well model. Therefore, although cognizant of Peaceman's model's (1983) restrictions, and that it is not the ideal model to be used with an unstructured grid [27, 41], it will be used in the EbFVM application in UTCHEM, as it is the base model used in UTCHEM for cartesian grids.

Since this model was originally obtained for cartesian grids, some changes are necessary in order to adapt this model for unstructured grids and to obtain reasonable solutions. Appendix A describes how Peaceman's model can be implemented for two-dimensional unstructured grids, according to the same procedure presented in Maliska *et al.* [37]. The extension for 3D grids is straightforward for vertical wells in grids that contain only horizontal layers. In this case, each layer of the grid can be considered as an areal 2D grid and the same approach in Maliska *et al.* [37] can be applied.

In order to run cases with more complex geometry or containing wells with arbitrary trajectories, a compatible model with these features would be necessary in order to obtain reliable results. If only the expected flow behavior is being analyzed, then the well model present in UTCHEM is sufficient.

## 3.8 Face mobility evaluation

In both discretized equations, (3.20) and (3.23), the only term that still needs to be approximated is the mobility term in the face. The technique commonly used for spatial interpolation of mobility terms is the upwind-type interpolation scheme. In this work a single-point upwind scheme is used, where the upstream weighting of the mobility in each face is determined analyzing whether the flux across the face is positive or negative [15], as respectively illustrated in Fig. 3.12.

In this example, if the flux across the face is positive, the property of node 1 is used as the interpolated variable. On the other hand, if the flux is negative, the property of node 4 is used in the face. Although this is the most simple upwind scheme, it provides physically consistent solutions, as the monotonicity is guaranteed [31]. However, it can provide a significant grid orientation effect, as it does not consider the actual flow direction into the numerical approximations of the advective terms. In order to avoid this
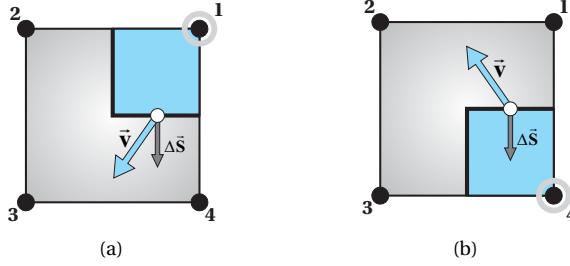
**Figure 3.12** – (a) Upstream node for positive flux (b) and negative flux.

undesirable effect, other schemes that take into account the true direction of the flow could be used [28].

## 3.9 Linear system assembling

In this section, the basic procedure for assembling the linear system for the pressure equation is presented. The procedure described here is also functional for the concentration equation when $\theta = n$ in Eq. (3.23), however the form of the equations in this case is different, as a Newton method is used in order to solve this equation implicitly. Therefore, for didactic purpose, the pressure equation, Eq. (3.20), will be considered here. Grouping the coefficients that multiply $P_1$, the global conservation equation for a given control volume associated to node $p$ can be written as

$$
\left( \frac{\phi_{\text{ref}} \, \Delta V \, C_t^{n-1}}{\Delta t} \right)_p (P_1)_p^n + \sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} \sum_{q \in \mathbb{N}^e} (\lambda_{Tc})_{\hat{f}}^{n-1} \left( b_q \right)_{\hat{f}} (P_1)_q^n =
$$

$$
\sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} \sum_{l=1}^{n_p} \sum_{q \in \mathbb{N}^e} (\lambda_{lc})_{\hat{f}}^{n-1} \left( b_q \right)_{\hat{f}} (w_l)_q^{n-1} + \tag{3.26}
$$

$$
\sum_{\kappa=1}^{n_{cv}} \left( Q_\kappa^{n*} \Delta V \right)_p + \left( \frac{\phi_{\text{ref}} \, \Delta V \, C_t^{n-1}}{\Delta t} \right)_p (P_1)_p^{n-1} \, ,
$$

where $q \in \mathbb{N}^e$ are all the nodes located at the vertices of element $e$ [40].

In the implementation of the EbFVM, it is common to perform the linear system assembling element by element, as the discretization method itself is based on local approximations for elements. However, as Eq. (3.26)
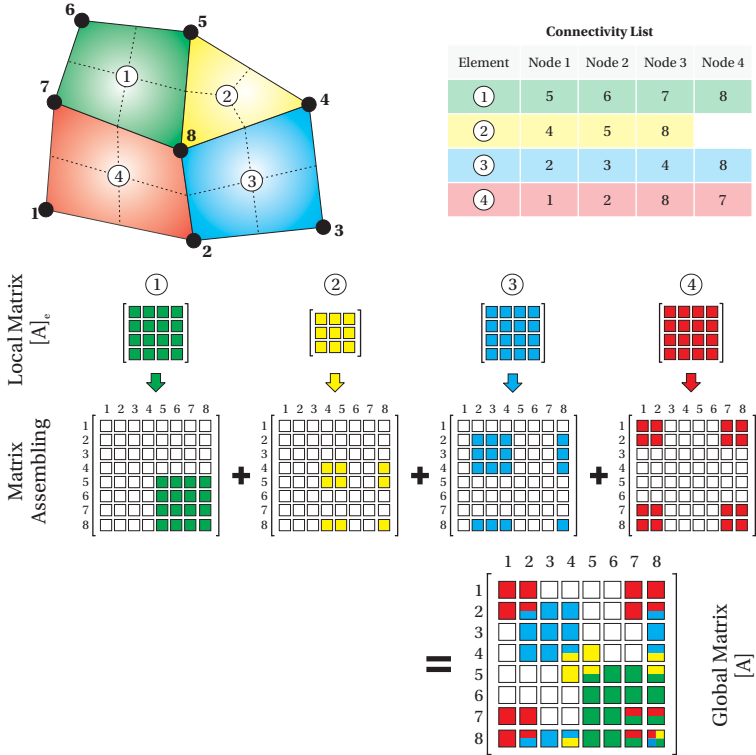
**Figure 3.13** – Matrix assembling example for a grid with four elements and eight nodes.

indicates, to realize the assembling for a given control volume, first the elements that are present in the formation of the control volume should be visited, then some faces in this element and then all nodes of this element. If it was performed this way, a given element would be visited several times, as it contributes to several control volumes.

Thus, it is more advantageous to visit each element only once at each time level, and calculate the contributions for each sub-control volume, storing these portions in a local matrix. After visiting all elements, the coefficients in the local matrix are transferred to the global matrix with the information in the connectivity list to find the correct position in the global matrix. The final form of the matrix is then assembled by summing the contributions of each element. Figure 3.13 illustrates the matrix assembling for a two-dimensional case containing one triangle and

three quadrilateral elements. As this grid has eight nodes, the dimension of the global matrix is $8 \times 8$.

In order to illustrate with more detail how the linear system is assembled, the local matrix of element 1 ($E_1$) will be obtained, and in the sequence, the equation of node 6 ($N_6$). As only element 1 contributes to the control volume associated with this node, with two faces involved in the mass balance equation, building the matrix of this element is enough for obtaining the equation of this control volume. Noting that each face contributes with two sub-control volumes in a 2D case, with a positive contribution for one control volume and a negative contribution for the other, one has in the local system:

- Sub-control volume 1 ($SCV_1$), associated with node 5 ($N_5$):

$$(A_{11})_{E_1} = \left( \frac{\phi_{\text{ref}} \, \Delta V \, C_t^{n-1}}{\Delta t} \right)_{SVC_1} + (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_1)_{\hat{f}_1} - (\lambda_{Tc})_{\hat{f}_4}^{n-1} (b_1)_{\hat{f}_4}$$

$$(A_{12})_{E_1} = (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_2)_{\hat{f}_1} - (\lambda_{Tc})_{\hat{f}_4}^{n-1} (b_2)_{\hat{f}_4}$$

$$(A_{13})_{E_1} = (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_3)_{\hat{f}_1} - (\lambda_{Tc})_{\hat{f}_4}^{n-1} (b_3)_{\hat{f}_4}$$

$$(A_{14})_{E_1} = (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_4)_{\hat{f}_1} - (\lambda_{Tc})_{\hat{f}_4}^{n-1} (b_4)_{\hat{f}_4}$$

- Sub-control volume 2 ($SCV_2$), associated with node 6 ($N_6$):

$$(A_{21})_{E_1} = (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_1)_{\hat{f}_2} - (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_1)_{\hat{f}_1}$$

$$(A_{22})_{E_1} = \left( \frac{\phi_{\text{ref}} \, \Delta V \, C_t^{n-1}}{\Delta t} \right)_{SVC_2} + (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_2)_{\hat{f}_2} - (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_2)_{\hat{f}_1}$$

$$(A_{23})_{E_1} = (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_3)_{\hat{f}_2} - (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_3)_{\hat{f}_1}$$

$$(A_{24})_{E_1} = (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_4)_{\hat{f}_2} - (\lambda_{Tc})_{\hat{f}_1}^{n-1} (b_4)_{\hat{f}_1}$$

- Sub-control volume 3 ($SCV_3$), associated with node 7 ($N_7$):

$$(A_{31})_{E_1} = (\lambda_{Tc})_{\hat{f}_3}^{n-1} (b_1)_{\hat{f}_3} - (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_1)_{\hat{f}_2}$$

$$(A_{32})_{E_1} = (\lambda_{Tc})_{\hat{f}_3}^{n-1} (b_2)_{\hat{f}_3} - (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_2)_{\hat{f}_2}$$

$$(A_{33})_{E_1} = \left( \frac{\phi_{\text{ref}} \, \Delta V \, C_t^{n-1}}{\Delta t} \right)_{SVC_3} + (\lambda_{Tc})_{\hat{f}_3}^{n-1} (b_3)_{\hat{f}_3} - (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_3)_{\hat{f}_2}$$

$$(A_{34})_{E_1} = (\lambda_{Tc})_{\hat{f}_3}^{n-1} (b_4)_{\hat{f}_3} - (\lambda_{Tc})_{\hat{f}_2}^{n-1} (b_4)_{\hat{f}_2}$$

- Sub-control volume 4 ($SCV_4$), associated with node 8 ($N_8$):

$$(A_{41})_{E_1} = (\lambda_{Tc})^{n-1}_{\dot{f}_4} (b_1)_{\dot{f}_4} - (\lambda_{Tc})^{n-1}_{\dot{f}_3} (b_1)_{\dot{f}_3}$$

$$(A_{42})_{E_1} = (\lambda_{Tc})^{n-1}_{\dot{f}_4} (b_2)_{\dot{f}_4} - (\lambda_{Tc})^{n-1}_{\dot{f}_3} (b_2)_{\dot{f}_3}$$

$$(A_{43})_{E_1} = (\lambda_{Tc})^{n-1}_{\dot{f}_4} (b_3)_{\dot{f}_4} - (\lambda_{Tc})^{n-1}_{\dot{f}_3} (b_3)_{\dot{f}_3}$$

$$(A_{44})_{E_1} = \left( \frac{\phi_{\text{ref}} \Delta V \, C_t^{n-1}}{\Delta t} \right)_{SVC_4} + (\lambda_{Tc})^{n-1}_{\dot{f}_4} (b_4)_{\dot{f}_4} - (\lambda_{Tc})^{n-1}_{\dot{f}_3} (b_4)_{\dot{f}_3}$$

In these equations, $\left( \phi_{\text{ref}} \Delta V \right)_{SVC_k}$ represents the porous volume of sub-control volume $k$, and the porous volume of the CV is the sum of the porous volume of the SCVs that form this CV.

The local matrix of the first element is built with the equations listed. The equation for node 6 can be directly assembled from line 2 of the matrix $[A]_{E_1}$, combining the independent term of Eq. (3.28). Therefore, for node 6 one can write

$$(A_{21})_{E_1} (P_1)^n_{N_5} + (A_{22})_{E_1} (P_1)^n_{N_6} + (A_{23})_{E_1} (P_1)^n_{N_7} + (A_{24})_{E_1} (P_1)^n_{N_8} = B_{N_6} \ , \quad (3.27)$$

where the independent term $B_{N_6}$ represents the RHS of Eq. (3.26). Writing Eq. (3.27) in a more compact way for any node $p$,

$$\sum_{k \in \mathbb{T}_p} A_k (P_1)^n_k = B_p \ , \tag{3.28}$$

where $k \in \mathbb{T}_p$ represents all nodes in the stencil of node $p$ [40]. This stencil is formed by the nodes from the elements that contain node $p$, that is, from the elements that are present in the control volume construction.

In Eq. (3.28), $A_k$ are the not-null coefficients from the discrete equation, associated to nodes $k$, and the independent term $B_p$ includes all terms in the equation of node $p$ that are not function of $P_1$ in the time level $n$. When all equations of all control volumes are grouped, a linear system with a sparse matrix is obtained. The finer the grid is, the higher the sparsity of the coefficient matrix [36].

# 4

# **TIMESTEPPING APPROACHES**

As presented in the integrated concentration equation, Eq. (3.23), it is necessary to choose the time level where the mobilities will be evaluated. Different formulations are obtained depending on this choice. In the case of $\theta = n-1$, an explicit evaluation of the terms will be conducted (IMPEC method). On the other hand, if $\theta = n$, a sequential semi-implicit formulation will be employed (SEQ method). This formulation is denominated semi-implicit because, in this work, in a given time step the pressure is always determined before the concentrations; thus, for the concentration equation the pressure is an explicit term. The fully implicit formulation, where both concentrations and pressure are determined implicitly and together in the same matrix, is not an available option in the UTCHEM simulator.

In the next sections the IMPEC and SEQ methods are presented.

## 4.1 IMPEC method

In this method, a complete explicit evaluation of the flux is performed, that is, $\theta = n$. From Eq. (3.23) is possible to rearrange the terms as

$$\left(\tilde{C}_\kappa\right)_p^n = \frac{\Delta t}{(\phi\Delta V)_p} \sum_{f\in\mathbb{F}_p} \sum_{l=1}^{n_p} \left[ (C_{\kappa l})_f^{n-1} (q_l)_f^{n-1} + (D_{\kappa l})_f^{n-1} \right] +$$
$$\frac{\Delta t}{(\phi\Delta V)_p} \left( \frac{R_\kappa^{n-1}}{\rho_\kappa} \right)_p + \left(\tilde{C}_\kappa\right)_p^{n-1} , \tag{4.1}$$

where all terms in the RHS are known.

In this scheme, first the pressure is solved implicitly through a linear system involving Eq. (3.27), using a time-lagged mobility, and then the concentrations are updated through a group of algebraic equations, represented by Eq. (4.1). Although the IMPEC method reduces drastically the computational effort in each time step, it imposes several restrictions on the magnitude of the timestep that can be employed for obtaining the concentrations without oscillations in the solutions [14]. The stability of the solution is governed by the Courant number (Cr), that is written in UTCHEM as

$$\text{Cr} = \Delta t \left( \frac{|Q_T|}{\phi\Delta V} \right)_{\max} , \tag{4.2}$$

where $Q_T$ is the total injection/production rate per wellblock, $\Delta t$ is the timestep, and $\phi\Delta V$ is the porous volume.

The simulator has three automatic options for determining the timestep size: selector based on the method of relative changes for the first three components (water, oil, and surfactant), selector based on the method of relative changes for all components, and selector based on changes in dimensionless concentration for all components, as described in UTCHEM-9.9 [56]. The Courant number calculated with the timestep from any of these options has to be between a minimum and a maximum value, specified by the user. In case either of these restrictions is not respected, the timestep is recalculated using the minimum/maximum Courant number, isolating $\Delta t$ from Eq. (4.2). In other works, studies were realized in order to verify the limitations of this condition, as well as to propose different forms to evaluate the stability criterion [14, 23].

One point that is appealing in the IMPEC strategy is the fact that only one linear system is solved in each time step, and the rest of the procedure is realized through algebraic expressions. Therefore, this is the

fastest approach on a pertimestep basis; however, the Courant stability criterion for an explicit scheme always requires a limited timestep size. Both the explicit updating of saturations or concentrations and the time-lagged and pressure-dependent mobility terms in the pressure equation are sources of instability. This instability can particularly arise in enhanced oil recovery simulations where shear-dependent viscosities and capillary number dependence of relative permeabilities exist [23]. The stability restrictions make the IMPEC method at times inefficient or even make it impossible to solve some simulation problems.

The following is the summary of the procedure for one time step applying the IMPEC algorithm:

1. Solve the pressure with the time-lagged mobilities.

2. Calculate the phase velocity in each face, applying Eq. (3.24).

3. Update the concentration field for all components, with Eq. (4.1).

4. Determine the new phase mobility terms.

5. Advance in time.

6. Determine the new timestep, based on the Courant number.

7. Return to step 1 until final time has reached.

Originally, the timestep adopted in the pressure and concentration equations are the same, usually being determined obeying the Courant stability condition. However, in certain problems the pressure field changes more slowly than the concentration field, and, in this way, it is possible to accelerate the algorithm calculating the pressure less times and, with the same total velocity field obtained with this pressure field, to update the concentration several times [28]. This method is an adaptation of the IMPEC method, described in the sequence, and will be designated as A-IMPEC.

## 4.2 Adaptive IMPEC method

Since the stability restriction in the IMPEC method is caused by the explicit approximation of the concentration, Hurtado *et al.* [28] proposed a scheme where the concentration would be updated with small timesteps during a certain period, obeying the stability criterion, but keeping frozen

the total velocity field during this period. Only after this the pressure linear system would be solved again and the new total velocity updated.

The strategy of using larger timesteps for the pressure was first proposed by Chen *et al.* [10] in a two-phase problem. However, in his work it was not clear how much larger this pressure timestep could be without damaging the numerical solution. In Hurtado *et al.* [28] an adaptive strategy for controlling the pressure timestep, based on the mean variation of the total velocity, was proposed for two-phase cases in two-dimensional domains. The scheme presented here is an extension for multiphasic three-dimensional cases and was applied for both the FDM and EbFVM methods. For the sake of simplicity, it will be described with more detail only for the formulation studied in the present work, but the idea is the same for both.

As the conventional IMPEC scheme, in a given time step $n$, first the pressure is solved using the time-lagged mobilities. After solving the pressure implicitly, the total velocity in each face, $(q_T)_{\hat{f}}$, is determined, using (3.24) for all phases[1]. The idea now is to solve the concentration for a certain number of steps, designated here by $m$ steps, keeping this total velocity stationary and determining the phase velocities in each step by applying the Buckley-Leverett expression, that is

$$(q_l)_{\hat{f}} = (F_l)_{\hat{f}} \left[ (q_T)_{\hat{f}} + \sum_{j=1}^{n_p} \lambda_{jc} [b]_{\hat{f}}^T \left( [\hat{w}_l]_e - [\hat{w}_j]_e \right) \right] , \qquad (4.3)$$

where the subscript $j$ is used in summing over the phases. The value $j = l$ can be included or not, as it does not contribute to the expression. The term $(F_l)_{\hat{f}}$ is the fractional flux of phase $l$, defined as

$$(F_l)_{\hat{f}} = \frac{\lambda_{lc}}{\displaystyle\sum_{j=1}^{n_p} \lambda_{jc}} . \qquad (4.4)$$

With the phase velocity, the concentration equation, Eq. (4.1), can be applied in order to update this variable. With the new concentrations, the mobilities can be updated as well as the phase velocities, applying this process until the concentration is solved $m$ times. Only after this period

---

[1]Actually, the fluxes are determined directly without calculating the velocities first, but it is more intuitive to say that the velocities are obtained instead.

the pressure, as well as the total velocity field, are solved again. This period where the total velocity is kept frozen, between the time level $n$ and $n+m$, is given by the pressure timestep ($\Delta t_P$), that is larger than the concentration timestep ($\Delta t_C$).

The concentration timestep is determined with the same criterion applied in the conventional IMPEC formulation, and for the pressure timestep an adaptive strategy is used, based on the variation of the total velocity field, as performed in Maliska *et al.* [42]. The idea is to allow large timesteps when this field changes slowly, reducing it progressively when the velocity starts to change faster. For this, the following relation is employed [42]:

$$\Delta t_P^{n+m} = \Delta t_P^n \frac{TOL}{\left(\Delta \tilde{q}_T\right)_{\max}^n} \ , \tag{4.5}$$

where $\Delta t_P^{n+m}$ is the new pressure timestep, after solving the concentrations $m$ times. The parameter *TOL* is the admissible variation for the parameter $\left(\Delta \tilde{q}_T\right)_{\max}^n$, which is related to the maximum variation of the total velocity, given by

$$\left(\Delta \tilde{q}_T\right)_{\max}^n = \frac{1}{Norm} \max_p \left( \frac{\sum_{f \in \mathbb{F}_p} \left| \left(q_T\right)_f^n - \left(q_T\right)_f^{n-1} \right|}{\phi_p \Delta V_p} \right) \ . \tag{4.6}$$

The numerator of the expression is the total flux variation in the control surface of the control volume associated to node $p$, and it is divided by the porous volume of the CV. This expression is calculated for all control volumes, in order to determine the maximum value in the grid. This maximum value is normalized with the parameter *Norm*, given by

$$Norm = \max_p \left( \frac{\sum_{f \in \mathbb{F}_p} \left| \left(q_T\right)_f^{n=1} \right|}{\phi_p \Delta V_p} \right) \ , \tag{4.7}$$

which is similar to the parameter defined before, but evaluated at the first time level, considering that $\left(q_T\right)_f^{n=0} = 0$ in all faces of all control volumes.

There is no physical limit for the maximum pressure timestep size that can be applied during a simulation. The constraint is related only to the error for keeping the same total velocity during a large period of time. Therefore, in this work the value specified for the tolerance *TOL* is 0.01,

which provided significant reductions in the computational time without introducing notable errors in the numerical solutions.

The following is the basic procedure for one pressure time step, from level $n$ to level $n+m$, applying the A-IMPEC algorithm:

1. Solve the pressure with the time-lagged mobilities.

2. Calculate the total velocity in each face, applying Eq. (3.24) for all phases.

3. Determine the new pressure timestep, using Eq. (4.5), and store the current time $t^{\text{old}} = t^n$.

4. Repeat the following $m$ times, until $t^{n+1} > t^{\text{old}} + \Delta t_P^{n+m}$:

   4.1. Calculate the phase velocity in each face, through Eq. (4.3).
   4.2. Update the concentration field for all components, with Eq. (4.1).
   4.3. Determine the new phase mobility terms.
   4.4. Advance the time by $t^{n+1} = t^n + \Delta t_C$.
   4.5. Determine the new concentration timestep, based on the Courant number.

5. Return to step 1 until final time has reached.

As will be seen in Chapter 5, this scheme reduces the computational time significantly, as the pressure linear system, usually the most consuming process in a simulation, is solved less times than the conventional IMPEC formulation. The algorithm can be applied to problems with any number of components and phases. The only limitation is when any compressibility is present. In this case the pressure, and, therefore, the total velocity, varies more in the same position when compared to an incompressible problem. Therefore, the pressure needs to be solved at each time step and the A-IMPEC approach cannot be used in this case.

## 4.3 Sequential method

The idea of the sequential method is to improve the stability of the IMPEC method by incorporating implicit treatment of concentrations, but without solving simultaneously for pressures and concentrations. Therefore, it consists basically of two steps. The first step is to obtain an implicit

pressure solution, exactly in the same manner as for the IMPEC strategy. The second step consists of an implicit solution for the concentrations, applying Newton's method and taking the derivatives of the mobility terms. In this way, from Eq. (3.23) with $\theta = n$, a non-linear system for the concentration is obtained, using the known pressure determined previously. This equation will be rewritten here, but substituting the phase flux into the conservation equation:

$$
\left[ \phi_{\text{ref}} \Delta V \frac{\left( \tilde{C}_\kappa^n - \tilde{C}_\kappa^{n-1} \right)}{\Delta t} \right]_p =
$$
$$
\sum_{f \in \mathbb{F}_p} \sum_{l=1}^{n_p} \left[ (C_{\kappa l})_f^n (\lambda_{lc})_f^n [b]_{\hat{f}}^T \left( \left[ \hat{P}_l \right]_e^n - \left[ \gamma_{\hat{l}} h \right]_e^n \right) + (D_{\kappa l})_f^n \right] + \left( \frac{R_\kappa^n}{\rho_\kappa} \right)_p .
\tag{4.8}
$$

The generated equation is non-linear, as the concentration is present implicitly in the mobility terms of the equation. Therefore, the application of Newton's method is viable in order to solve the non-linear system [15]. The implicit treatment of these variables yields more stability to the SEQ method than to the IMPEC method, creating the possibility to use larger timesteps during the simulation. This benefit is generally more advantageous than the use of the IMPEC method, even though some iterations in each time step are necessary.

Although it is applicable to any number of components and phases, in this work the SEQ method is implemented only for two-phase problems with two components (water/oil) for the EbFVM formulation. These restrictions are due to the difficulty in adapting a computational code that is written and prepared for the explicit evaluation of the concentrations to an implicit method. The sequential strategy needs a flash routine, that must be called along Newton's iterations, where all terms dependent on the concentration are evaluated. Such a routine was not implemented in the code; thus, it was not worth, for the present work, re-implementing all of the routines that are influenced by the concentrations, grouping them into the flash routine. For a case restricted to the components water and oil, the flash routine is much simpler, and, therefore, it was suitable to test the method with this simpler flow problem, deciding whether or not it is desirable to extend it for any number of components in future work.

Considering these two components, only the conservation equation

of one of them needs to be solved. The concentration of the other component can be determined from the restriction equation. Besides, only two phases are present in this case, with the aqueous phase ($l = 1$) composed only by component water ($\kappa = w$) and the oleic phase ($l = 2$) composed only by component oil ($\kappa = o$). That is,

$$
\begin{aligned}
C_{w1} &= 1 & C_{o1} &= 0 \\
C_{w2} &= 0 & C_{o2} &= 1 \ .
\end{aligned}
\tag{4.9}
$$

Choosing to solve the conservation equation of component water, the first step is to write Eq. (4.8) in terms of a residual function,

$$
(G_w)_p = \left[ \phi_{\text{ref}} \Delta V \frac{\left( \tilde{C}_w^n - \tilde{C}_w^{n-1} \right)}{\Delta t} \right]_p -
$$
$$
\sum_{f \in \mathbb{F}_p} \left[ (\lambda_{r1c})_{\hat{f}}^n [b]_{\hat{f}}^T \left( \left[ \hat{P}_1 \right]_e^n - \left[ \gamma \hat{1} h \right]_e^n \right) + (D_{w1})_{\hat{f}}^n \right] - \left( \frac{R_w^n}{\rho_w} \right)_p ,
\tag{4.10}
$$

where $(G_w)_p$ is the residue of the mass conservation equation for component water.

The objective of Newton's method is to reduce this residue, calculated in all control volumes, to a sufficiently small value. For this, expanding the residue relative to iteration $k+1$ in Taylor series, with second-order truncation error, results in

$$
G_w^{k+1} = G_w^k + \left. \frac{\partial G_w}{\partial \tilde{C}_w} \right|^k \Delta \tilde{C}_w \ .
\tag{4.11}
$$

By imposing $G_w^{k+1} = 0$ and omitting the superscript $k$, the residue can be written as

$$
-G_w = \frac{\partial G_w}{\partial \tilde{C}_w} \Delta \tilde{C}_w \ .
\tag{4.12}
$$

Now it is necessary to calculate the derivatives of the residual function, that will be part of the Jacobian matrix. In this work, only numerical derivatives were applied to the Jacobian assembling, in the form

$$
\frac{\partial G_w}{\partial \tilde{C}_w} = \frac{G_w \left( \tilde{C}_w + \Delta \tilde{C}_w \right) - G_w \left( \tilde{C}_w \right)}{\Delta \tilde{C}_w} \ ,
\tag{4.13}
$$

adopting $\Delta \tilde{C}_w = 10^{-4}$.

When substituting Eq. (4.10) into (4.12) and applying it for all control volumes, a linear system for iteration $k$ is obtained, in the form

$$[J]\{\Delta \tilde{C}_w\} = -\{G_w\} \ , \tag{4.14}$$

where $[J]$ is the Jacobian matrix, containing all partial derivatives of the residues applied in all control volumes, $\{\Delta \tilde{C}_w\}$ is the vector with the corrections for the water concentration, and $\{G_w\}$ is the vector containing the residues referent to the available concentration values. The linear system here is assembled in the same fashion as explained in the previous chapter for the pressure equation.

After solving this linear system, it is possible to update the water and oil concentration by

$$\left\{ \begin{array}{l} \tilde{C}_w^{k+1} = \tilde{C}_w^k + \Delta \tilde{C}_w \\ \tilde{C}_o^{k+1} = 1 - \tilde{C}_w^{k+1} \ . \end{array} \right. \tag{4.15}$$

With the new concentration fields, the mobilities are updated in the flash routine, and the iterative process is repeated, until $G_w = 0$ or $\Delta \tilde{C}_w = 0$, given a certain tolerance. In this work, only the second option, based on the maximum $\Delta \tilde{C}_w$ encountered in the grid, is checked, with the tolerance equal to $10^{-3}$. After this, the next time step is achieved.

The procedure for one time step can be summarized in the following:

1. Solve the pressure with the time-lagged mobilities.

2. Calculate the total velocity in each face, applying Eq. (3.24) for all phases.

3. Repeat the following until convergence:

   3.1. With the available values for the concentrations, calculate the mobility terms in the flash routine.

   3.2. Calculate the phase velocity in each face, through Eq. (4.3).

   3.3. Compute the residue of the water equation in all control volumes, with Eq. (4.10).

   3.4. Compute the partial derivatives of the residual functions in all control volumes, obtained numerically through Eq. (4.13). In order to evaluate the term $G_w(\tilde{C}_w + \Delta \tilde{C}_w)$, the flash routine

       must be called again, with the provided increment in the concentration.

    3.5. Solve the linear system given in Eq. (4.14).

    3.6. Update the concentrations with the correction from the solution of the linear system, using Eq. (4.15).

4. Advance in time.

5. Determine the new timestep, in the same way as for the pressure timestep in the A-IMPEC method.

6. Return to step 1 until final time has reached.

After achieving convergence for the component concentrations, the time can be advanced with the new timestep. There are several ways for determining the timestep in the SEQ approach. In this work, the same strategy applied for the pressure timestep in the A-IMPEC method is used for both the concentration and the pressure, so that $\Delta t_C^{\text{SEQ}} = \Delta t_P^{\text{SEQ}} = \Delta t_P^{\text{A-IMPEC}}$. Although there is less restriction in the SEQ method, since it is more stable than the IMPEC method, some attention is necessary in the timestep size in order to produce solutions with the desirable precision. This is due to the approximation in the time derivative as well as to the decoupling between pressure and concentration in this method.

CHAPTER

# 5

# **RESULTS**

In this chapter some application examples employing the formulation described in previously chapters are presented. The examples were chosen to demonstrate several aspects of the formulation could be evaluated. They will be presented according to the following division:

- Section 5.1: The EbFVM formulation is validated using a diffusion problem with known analytical solution.

- Section 5.2: A comparison between the EbFVM and the original formulation (FDM) is performed for single-phase, two-phase, and three-phase flow problems.

- Section 5.3: The computational time is analyzed for the different timestepping approaches presented in Chapter 4.

- Section 5.4: A case with a more complex 3D geometry is presented, where the potentiality of the method is clearly evidenced.

## 5.1 EbFVM validation

In this section, the implementation of all element types is validated by comparing the numerical solution for a diffusion problem with its analytical solution. The problem considered here corresponds to an example proposed for comparing different discretization methods, as used

in Maliska *et al.* [41] and LATP [33]. It considers a homogeneous and isotropic medium, described by the following diffusion equation:

$$-\vec{\nabla}^2\Phi = R \quad \text{in } \Omega \, , \tag{5.1}$$

where $\Phi$ is a generic variable, which could be the aqueous phase pressure $P_1$ in the case of UTCHEM, and $R$ is the source term, that can be the well injection/production. The domain $\Omega$ is a cubic box defined by $0 \leq x \leq 1$, $0 \leq y \leq 1$ and $0 \leq z \leq 1$.

In this problem, the source term considered is dependent on the position, given by

$$R(x, y, z) = 12\pi^2 \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \, . \tag{5.2}$$

This way, the analytical solution is given by

$$\Phi(x, y, z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \, . \tag{5.3}$$

In order to make UTCHEM able to simulate such a problem, the pressure equation, Eq. (2.8), can be reduced to Eq. (5.1) by using only one component and setting the porosity, absolute permeability, and mobility to 1 and setting the total compressibility, capillary pressure, and gravity to 0.

Several meshes were considered for solving this problem, in order to test the formulation with different element types and different levels of refinement. All four type of elements considered in the formulation are tested here, with meshes that include only one type of element at a time. Hybrid grids are not considered in this test due to the difficulty in generating them automatically. Grids containing tetrahedron, prisms, and pyramids were generated based on the hexahedron element, that is, each hexahedron was divided into six tetrahedrons, two prisms or six pyramids [41], as illustrated in Fig. 5.1. Note that the grid containing pyramids requires an additional node in the middle of the hexahedron, thus the number of nodes in this grid is higher than in the grids composed by hexahedrons, tetrahedrons, or prisms.

Both regular elements, in equally spaced grids, and distorted elements, providing non-equally spaced grids are considered. The latter were obtained by moving the grid nodes randomly. The purpose of using these grids is to determine to what extent the irregularities affect the
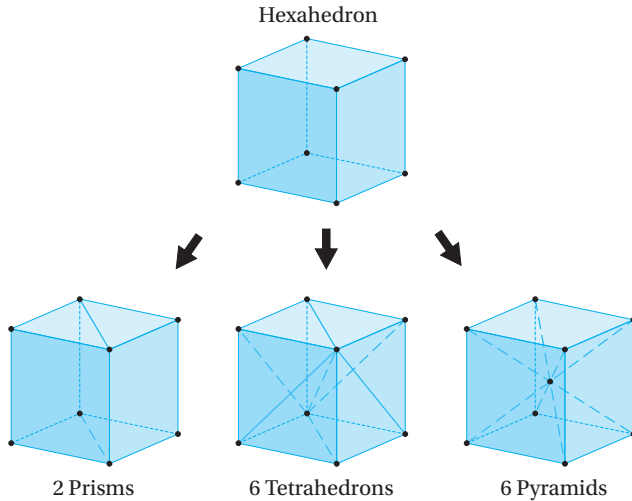
**Figure 5.1** – Subdivision of the hexahedron in different elements.

convergence rate of the numerical solutions. In order to determine the convergence order, progressive refinements were considered for each type of grid, with the coarsest grid size composed by $8 \times 8 \times 8$ elements and the finest grid by $100 \times 100 \times 100$ elements. Note that even for grids composed by tetrahedrons, prisms, or pyramids, the grid size is based on the number of hexahedrons before the element division. Figure 5.2 illustrates the regular and distorted grids for the coarsest option.

After obtaining the numerical solution in each grid, the average values of the solution error were calculated. As performed in [41], the L2-norm of the numerical solution error is computed using the following expression:

$$\varepsilon = \left\{ \frac{\sum\limits_{p} \left[ \Phi(x_p, y_p, z_p) - \hat{\Phi}_p \right]^2 \Delta V_p}{\sum\limits_{p} \left[ \Phi(x_p, y_p, z_p) \right]^2 \Delta V_p} \right\} \; , \tag{5.4}$$

where the summations run over all grid nodes, $\Phi(x_p, y_p, z_p)$ is the analytical solution evaluated at the node coordinates, and $\hat{\Phi}_p$ is the numerical solution obtained for node $p$.

In the sequence, the convergence order is presented for the regular and distorted grids considered. The graphics represent the variation of the error as the grids are refined, with the horizontal axis corresponding
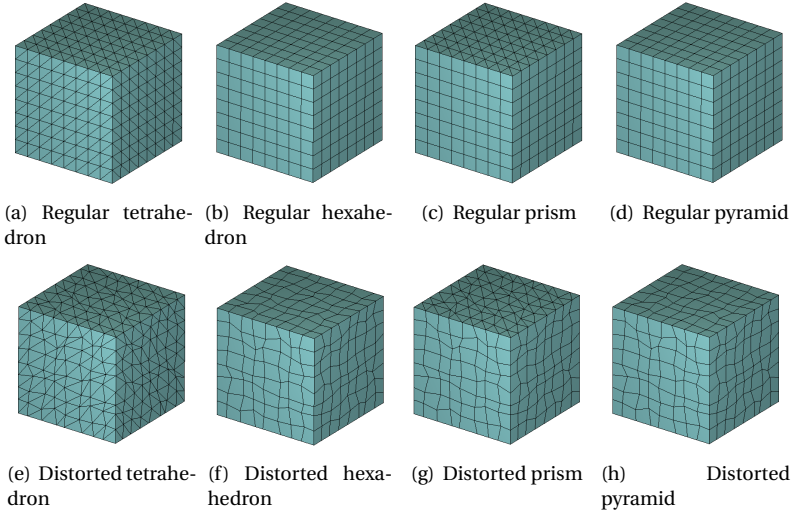
(a) Regular tetrahe-dron  (b) Regular hexahe-dron  (c) Regular prism  (d) Regular pyramid

(e) Distorted tetrahe-dron  (f) Distorted hexa-hedron  (g) Distorted prism  (h) Distorted pyramid

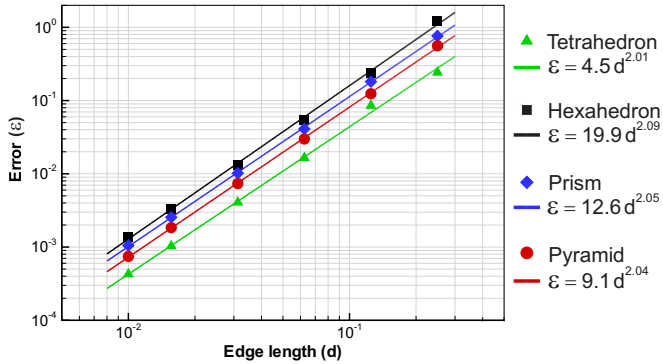**Figure 5.2** – Regular and distorted grids used in the validation problem.

to the average size $d$ of the grid elements, that is the edge length of the corresponding regular hexahedron.

As expected, the convergence rate for variable $\Phi$ is of second order for all element types. Tetrahedrons provided the smallest error, while hexahedrons provided the highest error, both in the regular and distorted cases. For this problem, the element distortion produced only a very slight increase in the errors, except for the pyramid, in which case it decreased. The trendlines are practically the same as well, and therefore, the distortion has a minor effect on the error norm.

This test does not guarantee that the implementation of the EbFVM in UTCHEM is free of errors, as it did not account for anisotropic and het-erogeneous medium, as wells as the advective term. However, it provides a very good indication that at least the geometrical variables, as shape functions, face areas, etc., are correctly implemented. In order to verify features not included in this test, the next section compares the solutions from the EbFVM and the original FDM method, which was extensively tested and validated by several ways, as presented in UTCHEM-9.0 [55].
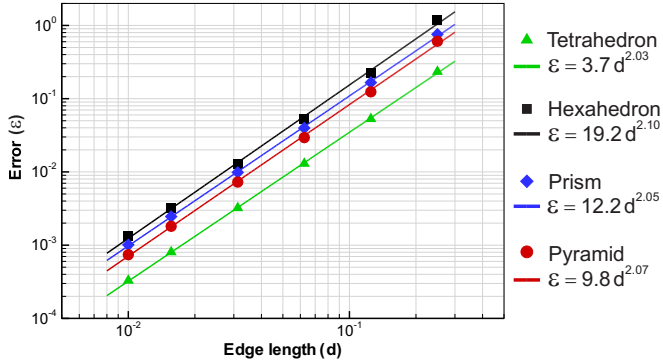
(a) Regular elements



(b) Distorted elements

**Figure 5.3** – Error in the variable $\Phi$ for regular and distorted grids.

## 5.2 FDM vs. EbFVM

As stated before, in this section the solution obtained by employing the new implementation (EbFVM) is compared to the solution obtained with the original formulation present in the simulator, which employs the traditional finite difference method (FDM). Three cases were chosen in order to make this comparison:

- Case 1: Single-phase problem (only water).

- Case 2: Two-phase problem (water and oil).

- Case 3: Three-phase problem (water, oil, and microemulsion).

For the sake of simplicity, no chemical reactions, no adsorptions, and zero capillary pressure are assumed. These effects do not depend on the method used to discretize the equations as they do not require any geometric information. They were neglected for the purpose of model verification.

In all tests, a Cartesian grid composed of $45 \times 45 \times 5$ gridblocks applying the FDM method was used as the reference solution. Three grids were used in the EbFVM method: two of them composed only by hexahedrons and one hybrid grid, composed by all four types of elements. The domains contain one injection and one production well, and they can be visualized in Fig. 5.4.



(a) Cartesian - $45 \times 45 \times 5$ gridblocks

(b) Hexahedron - $15 \times 15 \times 5$ elements

(c) Hexahedron - $45 \times 45 \times 5$ elements

(d) Hybrid - 6647 elements
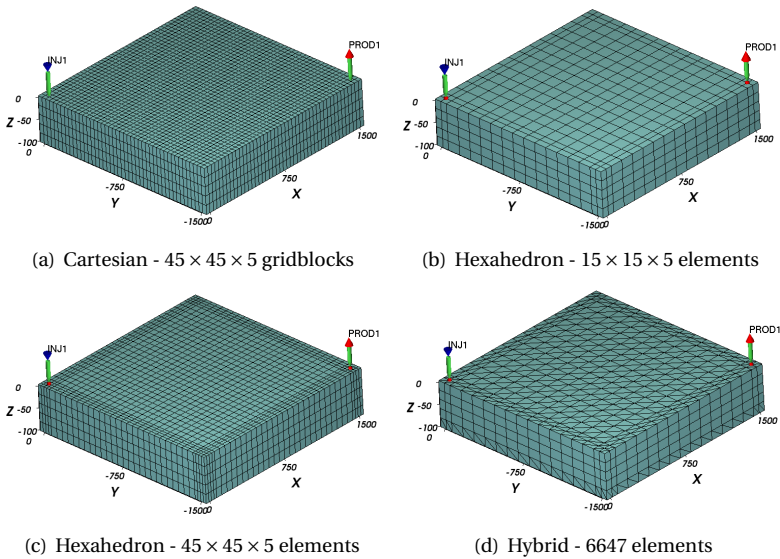
**Figure 5.4** – Grids used in the comparison between FDM and EbFVM.

It is important to note that the wells in the FDM method are located in the center of the gridblocks, while in the EbFVM the wells are defined in the grid nodes. Therefore, in order to place the wells in the same physical position in both methods, a refinement in the corners of the grids used in the EbFVM is necessary, as seen in Fig. 5.4. Furthermore, although this method is capable of dealing with complex geometries and unstructured grids, these simple structured grids are being used here only

for the purpose of comparison. Moreover, the method will treat them as unstructured grids even though they are structured.

The elements in the hybrid grid are disposed in the five layers as illustrated in Fig. 5.5. The top of the first layer is composed of prisms. The second and fourth layers are composed of both tetrahedrons and pyramids. The third layer is composed by hexahedrons and, finally, the fifth layer is composed only of tetrahedrons.
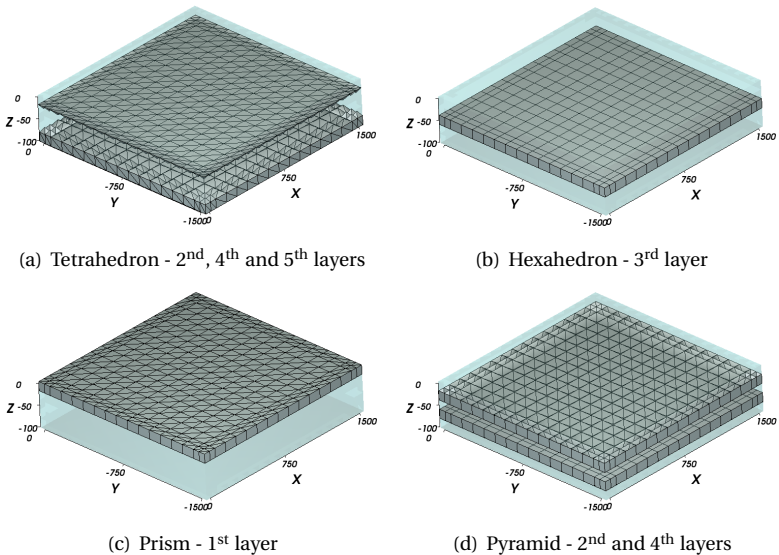


(a) Tetrahedron - 2$^{nd}$, 4$^{th}$ and 5$^{th}$ layers      (b) Hexahedron - 3$^{rd}$ layer

(c) Prism - 1$^{st}$ layer      (d) Pyramid - 2$^{nd}$ and 4$^{th}$ layers

**Figure 5.5** – Different elements in the hybrid grid.

## 5.2.1 Case 1 − Single-phase

For the single-phase problem, only component water is present. The input data for this case is shown in Table 5.1. It is important to note in this table that in both wells the condition is the prescribed bottom-hole pressure (BHP). Therefore, the well flow rate is an unknown of the problem, and will be determined by applying Eq. (3.25), which relates the flow rate to the pressure gradient between the reservoir and the well through the use of a well index. The flow rate will be correct only if both the well index and the reservoir pressure are well-calculated.

**Table 5.1** – Input data for Case 1.

| | | |
|---|---|---|
| | Reservoir domain | 1500 x 1500 x 100 ft |
| | Initial saturation | 100% water |
| | Initial pressure | 1000 psi |
| **Rock** | Porosity | 20% |
| | Permeability X | 100 mD |
| | Permeability Y | 100 mD |
| | Permeability Z | 10 mD |
| | Compressibility | $0\ \text{psi}^{-1}$ |
| **Fluid** | Water viscosity | 1 cP |
| | Water specific weight | 0.43 psi/ft |
| | Water compressibility | $0\ \text{psi}^{-1}$ |
| **Wells** | Injected fluid | 100% water |
| | Injection BHP | 5000 psi |
| | Production BHP | 500 psi |

An observation must be made regarding the units used in the tests performed in this work. They are a mixture of common units used in reservoir applications, such as millidarcy and centipoise, and units from the English system. The use of reservoir units is commonsense in problems involving fluid flow in porous media, and, if converted to the International System of Units, would provide confusing values for the user that is used to using these units. Regarding the other variables expressed in the English system, although the simulator can input values in the International System, it will always convert them to the English system of units, exporting the results in such a system. The conversion factors to International System are exposed in the beginning of this dissertation.

As this case test considers incompressible fluids and rock, it is a steady-state problem, and, therefore, there is only one value for the well flow rate during the simulation for each grid considered, as shown in Table 5.2. The largest difference in the results is less than 1%; thus, the solutions are in good agreement and it is possible to say that the well model, responsible for calculating the well index, is applicable to this type of grid.

As stated in Chapter 3, the application of the modified Peaceman's model is possible only because the reservoir contains regular horizontal layers and the wells are perpendicular to these layers, in the vertical direction.

**Table 5.2** – Well flow rates.

| Grid | Flow rate [ft³/d] | Difference |
|------|-------------------|------------|
| Cartesian - 45x45x5 | 57725 | - |
| Hexahedron - 15x15x5 | 58194 | 0.81% |
| Hexahedron - 45x45x5 | 57938 | 0.37% |
| Hybrid - 6647 elements | 57773 | 0.08% |

For illustrative purpose, Fig. 5.6 shows the pressure field obtained for this case, which is very similar for all grids considered.
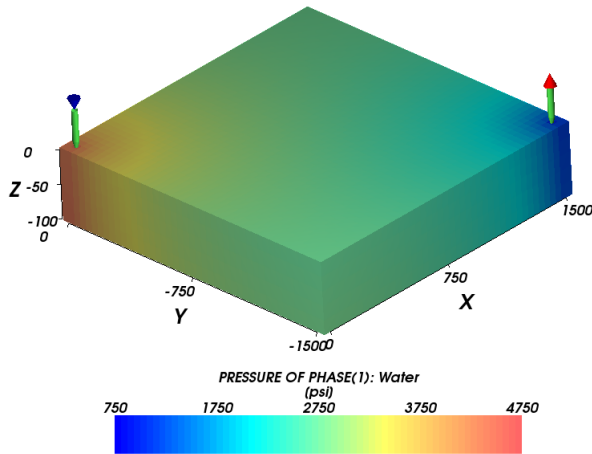


**Figure 5.6** – Pressure field for Case 1.

## 5.2.2 Case 2 – Water-oil flooding

Now, a two-phase system is considered, with the aqueous phase composed only by component water and the oleic phase only by component oil. Therefore, the phases, as well as the components, will be designated as water and oil. Table 5.3 illustrates the input data for this case.

**Table 5.3** – Input data for Case 2.

| | | |
|---|---|---|
| | Reservoir domain | 1500 x 1500 x 100 ft |
| | Initial saturation | 10% water - 90%oil |
| | Initial pressure | 1000 psi |
| **Rock** | Porosity | 20% |
| | Permeability X | Variable per layer |
| | Permeability Y | Variable per layer |
| | Permeability Z | 10 mD |
| | Compressibility | $5 \times 10^{-6}\, psi^{-1}$ |
| | Reference pressure | 14.7 psi |
| **Fluid** | Water viscosity | 1 cP |
| | Water specific weight | 0.43 psi/ft |
| | Water compressibility | $0\, psi^{-1}$ |
| | Oil viscosity | 10 cP |
| | Oil specific weight | 0.29 psi/ft |
| | Oil compressibility | $0\, psi^{-1}$ |
| **Wells** | Injected fluid | 100% water |
| | Injection flow rate | $40000\, ft^3/d$ |
| | Production BHP | 500 psi |

Here, the rock is compressible and the injection well condition is the prescribed total flow rate. Also, the absolute permeabilities in the horizontal directions are equal but variable per layer, as depicted in Fig. 5.7. Another important variable that is not present in Table 5.3 is the phase relative permeability. For this case, a very simple condition is used, as shown in Fig. 5.8.

The simulation time for this flow problem was 1500 days, which provided 1.33 pore volumes injected (PVI)[1]. The timestepping approach used was the traditional IMPEC method, with a constant timestep, equal

---

[1]PVI is a dimensionless time commonly used in reservoir simulation, which relates the injection flow rate, the time and the reservoir total porous volume.
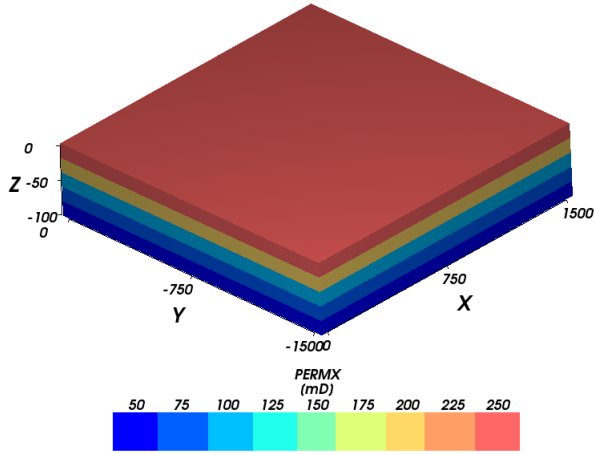
**Figure 5.7** – Permeability field for Case 2.
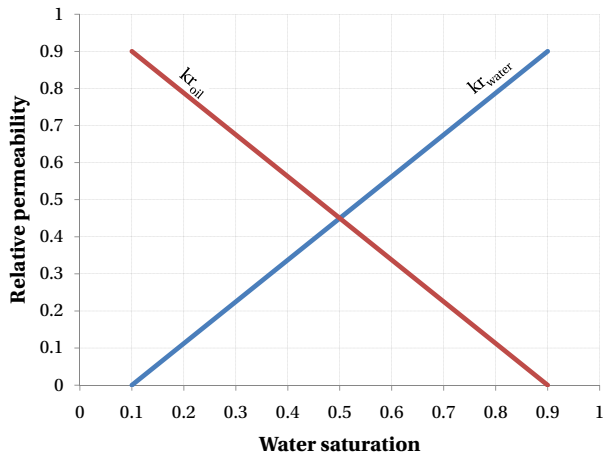


**Figure 5.8** – Water and oil relative permeabilities for Case 2.

to 0.05 day for all grids. The maximum Courant number obtained with this $\Delta t$ was 0.34 in the finer grid, which provides stable solutions for water-oil flow problems.

Figure 5.9 illustrates the cumulative oil recovery[2], the oil cut[3] and the total production rate for this case, where it is possible to note that the solutions are in good agreement with the reference FDM solution.


## 5.2.3  Case 3 – Chemical flooding

For this case two other components are present in the fluid flow: surfactant and polymer. Consequently, another phase will be present along the simulation, called microemulsion, which is a thermodynamically stable mixture of water, surfactant, and one or more organic components [55]. The basic input data is the same as used in Case 2, including the data related to timestepping. The only difference is in the injected components, where the aqueous phase injected contains 97% of water, 3% of surfactant, and 2000 ppm of polymer. Other parameters related to these two new components and to the phase behavior were based on an example available in UTCHEM-9.0 [55], and will not be discussed here.

The production well curves are illustrated in Fig. 5.10, where again a good agreement between the solutions is obtained. Comparing the water-oil and chemical flooding results, it is clear that when surfactant and polymer are injected, the oil recovery is remarkably enhanced, with a much sharper fluid front. Enhanced oil recovery processes, such as the one considered here, have utilized surfactants as they create low interfacial tension in order to reduce capillary forces and thus mobilize trapped oil, whereas polymers are used to reduce fluid mobility improving the sweep efficiency of the reservoir (*i.e.*, increasing the volume of the permeable medium contacted at any given time) [32, 51].

have utilized surfactants as they create low interfacial tension in order to reduce capillary forces and thus mobilize trapped oil, whereas polymers are used to reduce fluid mobility improving the sweep efficiency of the reservoir (i.e., increasing the volume of the permeable medium contacted at any given time) [32, 51].

---

[2]Cumulative oil recovery is the cumulative volume fraction of oil produced along the simulation.
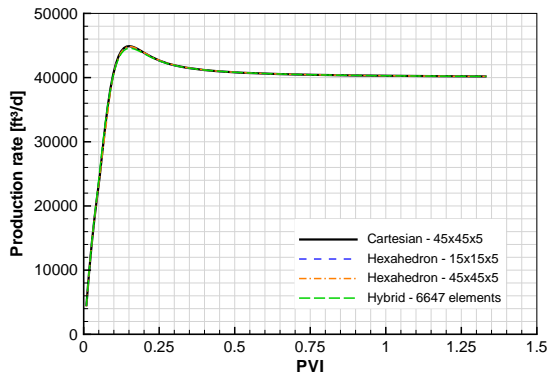
[3]Oil cut is defined as the oil production divided by the total production.
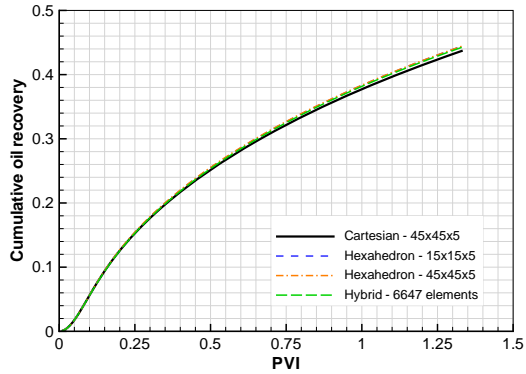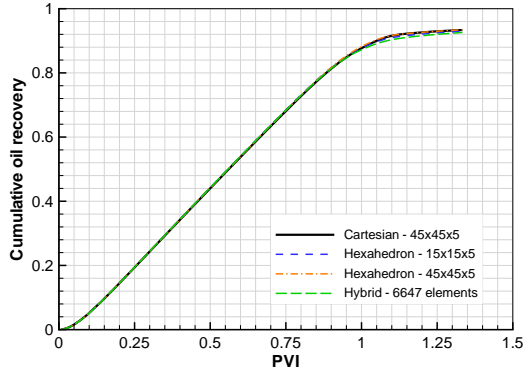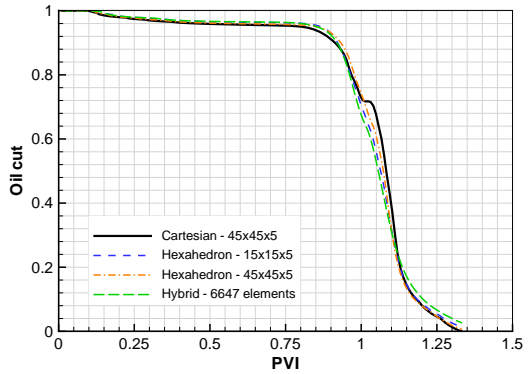
(a) Cumulative oil recovery
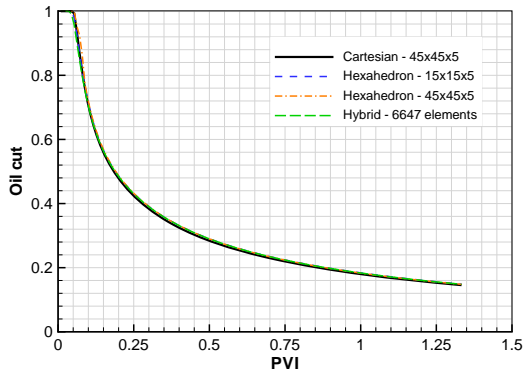


(b) Oil cut



(c) Total production rate

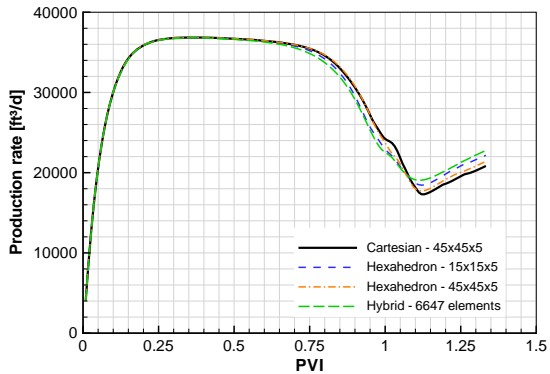**Figure 5.9** – Production well curves for Case 2.

(a) Cumulative oil recovery



(b) Oil cut



(c) Total production rate

**Figure 5.10** – Production well curves for Case 3.

## 5.3  Timestepping approaches comparison

In this section, the timestepping approaches described in Chapter 4 are applied, presenting the computational time for each approach in two test cases: a two-phase and a three-phase problem, which are very similar to the test cases applied in Section 5.2. The main differences are that here the rock is incompressible and that the absolute horizontal permeabilities are constant and equal to 100 mD, instead of variable per layer. In addition, the grids used now are equally spaced, instead of finer in the corners, and the vertical wells are located in the corners of the domain, as in the five-spot well pattern. The reservoir domain, however, is the same.

In the first case the IMPEC, A-IMPEC and SEQ approaches are applied only for the EbFVM. For the chemical flooding case, the SEQ method is not used, as it is implemented only for water-oil flow problems. In this case, the IMPEC and A-IMPEC are applied for both FDM and EbFVM formulations.

### 5.3.1  Case 1 – Water-oil flooding

In order to compare the CPU time for different grid refinements, the following grid sizes are used in this case: $5 \times 5 \times 5$, $15 \times 15 \times 5$, $45 \times 45 \times 5$, and $75 \times 75 \times 5$.

The traditional IMPEC formulation uses the same timestep for both the concentration and pressure equations. In all grids the minimum and maximum Courant number accepted are 0.1 and 0.6, respectively, which provided in the finer grid, for example, a minimum timestep equal to 0.004 and a maximum equal to 0.019. The A-IMPEC method uses different timesteps for each equation, with the same $\Delta t$ from the IMPEC method for the concentration and a larger timestep for the pressure equation. On the other hand, the SEQ method uses the same but a higher timestep for both equations.

The variation of the pressure timestep for the A-IMPEC and for the timestep in the SEQ method along the simulation for the the finer grid are illustrated in Fig. 5.11. The curves for the other grid sizes are very similar, keeping the same behavior and changing only slightly in the timestep magnitudes.

The oscillation between 0.05 and 0.25 PVI in the pressure timestep present in the A-IMPEC is due to the front of water that is reaching the production well.  When this happens, the total velocity changes more
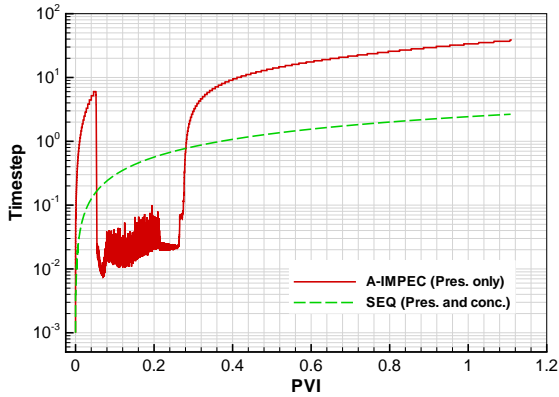
**Figure 5.11** – Timestep along the simulation for the grid $75 \times 75 \times 5$ in Case
1.

rapidly, and the method tries to increase the timestep but the total velocity
variation bars this increase in the next step by reducing it again. After a
larger volume of water has been produced, the total velocity field stabilizes
and the pressure timestep can be increased without restrictions.

While the concentration equation advances with a maximum of 0.019
day, the pressure equation is solved in each 35-day period in the end of
the simulation. For the SEQ method, a smaller timestep is applied in the
pressure equation, but for the concentration it is much larger than the one
applied in the A-IMPEC, reaching 2.6 days in the final time.

In order to verify that the solutions obtained with these different
approaches are correct, the oil cut curve for the finer grid is compared to
the one obtained with the traditional IMPEC. This comparison is shown
in Fig. 5.12, where a good agreement between the solutions is present.
Therefore, the timesteps used in both the A-IMPEC and SEQ methods are
adequate in this test case.

In the sequence, the CPU time spent for solving the pressure equation
and the concentration equation as well as for the total simulation is
presented. The comparison is made for different grid sizes using the three
timestepping approaches described, as shown in Fig. 5.13.

Analyzing first the pressure equation, where the linear system is
assembled and then solved, it is possible to identify that both curves for the
IMPEC and A-IMPEC have almost the same inclination, but with a smaller
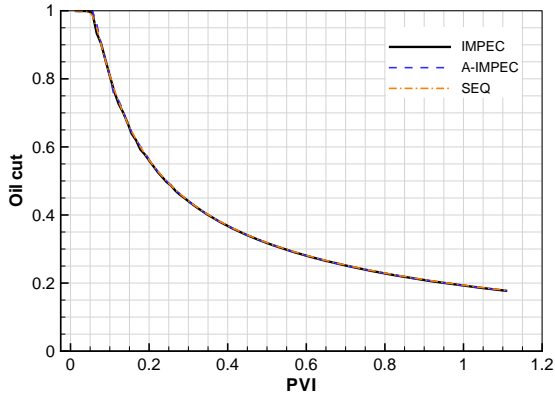magnitude for the adaptive option, being more than three times faster. The

**Figure 5.12** – Oil cut for the grid $75 \times 75 \times 5$ in Case 1.

sequential method spends more time for coarser grids but it provides much better computational times when the grid becomes finer.

For the concentration equation, the IMPEC and A-IMPEC provide the same CPU time, as these methods use the same concentration timestep for solving it. Although the SEQ method uses larger timesteps for the concentration equation, it needs some iterations in each step, due to the non-linearities. For the grids analyzed, this resulted in additional costs, but for even finer grids this method would be faster than the other two methods, as its curve inclination is smaller.

Finally, the total simulation time is the sum of the time spent in the pressure and concentration equations as well as the time consumed in auxiliary routines. For both the IMPEC and A-IMPEC methods, the most consuming process is solving the pressure equation, especially for the former; thus, the total time curve has a similar behavior to the pressure equation curve. On the other hand, for the SEQ method, the time spent in the pressure and concentration equations is of a similar order and magnitude, and the total time curve inclination is in between these two curves, presenting meaningful reductions in the computational time for refined grids. Therefore, it is a good indication for real reservoir problems with more than two components, being a good topic of study for future work.
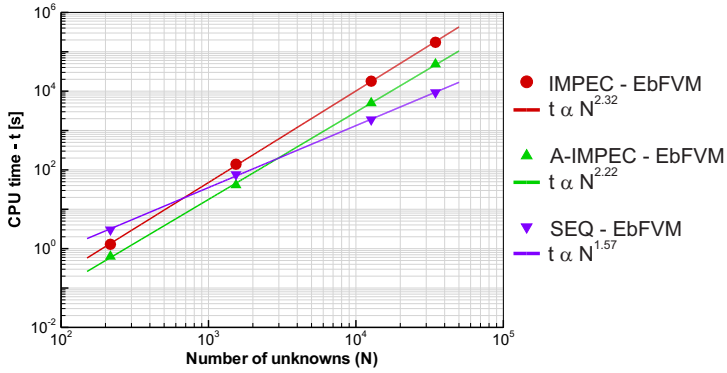
(a) Pressure equation
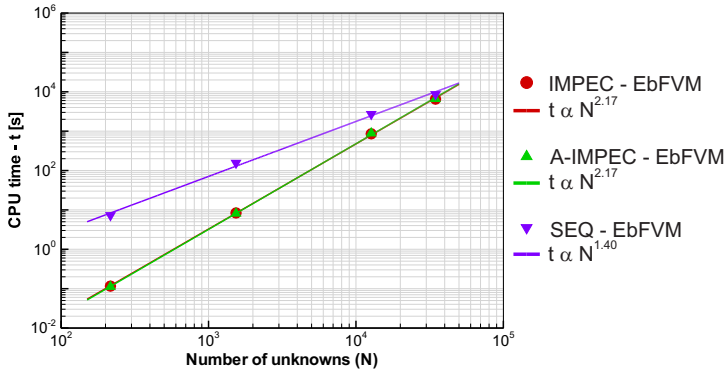


(b) Concentration equation
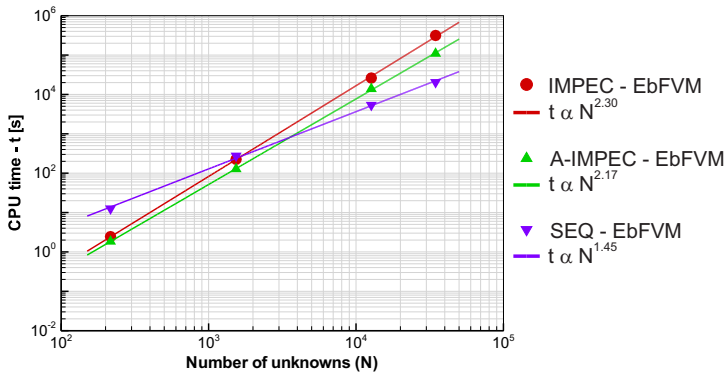


(c) Total time

**Figure 5.13** – CPU time for different routines in Case 1.

## 5.3.2 Case 2 – Chemical flooding

This case has the same input data of Case 1; however, 3% of surfactant and 2000 ppm of polymer are injected. The finest grid used in the first case is not applied here, as the presence of chemical components requires more computational effort, and it would take too long to be solved. In addition, the maximum Courant number accepted was set to 0.1 instead of 0.6, as the problem became more unstable when surfactant and polymer were injected. Therefore, the finest grid now is $45 \times 45 \times 5$ elements.

In this case, the SEQ method was not used as it was not implemented for chemical flooding processes; thus, only the traditional method and its adaptive option will be compared. The variation of the pressure timestep for the A-IMPEC along the simulation using the EbFVM method for the $45 \times 45 \times 5$ grid is illustrated in Fig. 5.14. As the front of fluid is sharper when chemical components are injected, the water reaches the production well later than in the water-oil flow problem. Thus, the total velocity stays stable for a longer period, providing larger timesteps during that period. This will result in less time spent in the pressure equation compared to the time spent in Case 1, as seen in the sequence.
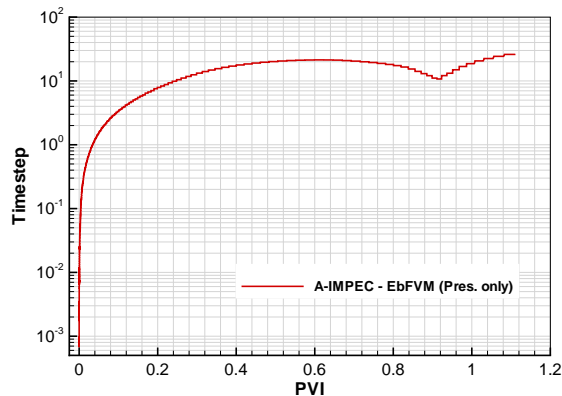


**Figure 5.14** – Timestep along the simulation for the grid $45 \times 45 \times 5$ in Case 2.

Similar to Case 1, the oil cut in the production well is presented in the sequence in order to verify whether or not the timesteps used are adequate for this problem. Figure 5.15 compares the solutions for the FDM

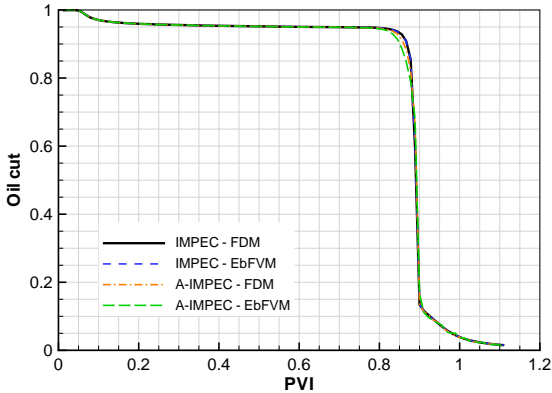and EbFVM using both timestepping approaches, considering the original formulations as the reference solution.



**Figure 5.15** – Oil cut for the grid $45 \times 45 \times 5$ in Case 2.

After verifying that the solutions are in good agreement, the computational time is compared in Fig. 5.16, in the same way as in the previous case.

For the pressure equation, comparing the CPU time for cases 1 and 2, the use of the traditional IMPEC method required more CPU time in the first case, due to the additional microemulsion phase. On the other hand, when using the adaptive option, although an additional phase is present, the second case was simulated even faster, due to the pressure timesteps used here, illustrated previously in Fig. 5.14. Therefore, contrary to Case 1, the inclination of the curves for the adaptive option is much smaller than those for the traditional method.

Furthermore, in this case the concentration equation is the most restrictive process when the A-IMPEC algorithm is employed. Therefore, as it does not provide any reduction in the time consumed for solving the concentrations, the time-saving during the whole simulation is not as significant as the time-saving in the pressure equation.

It is also possible to note that the EbFVM spends more time than the FDM formulation, as it is a scheme that uses more points than the latter. In the concentration equation, this difference is not as large as in the pressure equation. This is mostly due to the assembling of the pressure linear system, which demands extra time in the EbFVM formulation. This

process is not optimized in the current implementation, and it needs to be addressed in future work.
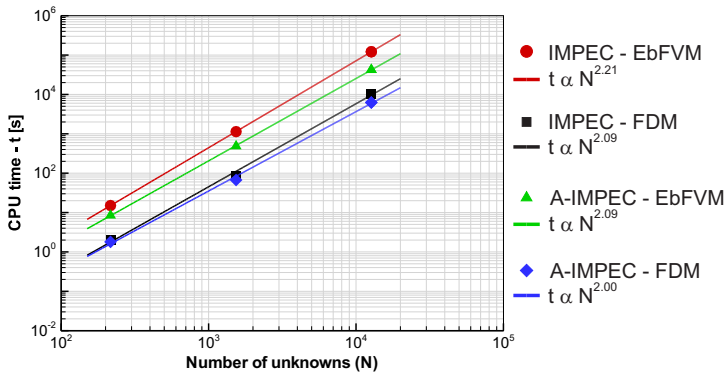
From Fig. 5.16 it is clear that the A-IMPEC algorithm is a good option for saving computational time when incompressible problems are simulated.
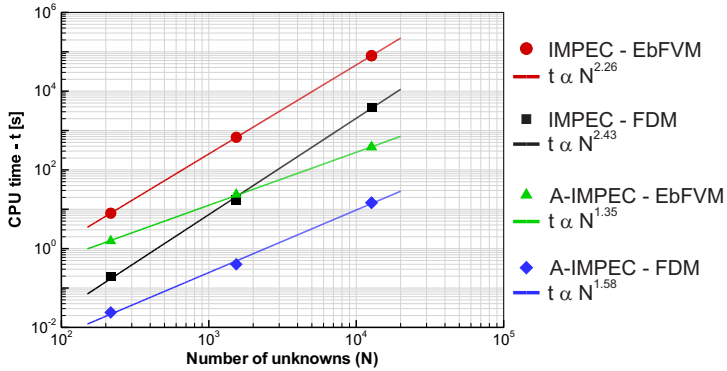
(a) Pressure equation
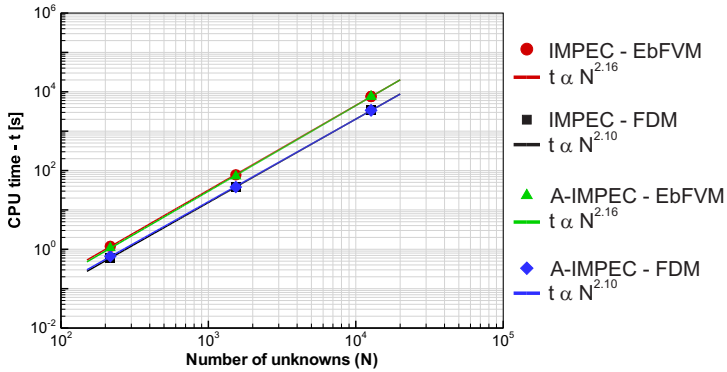


(b) Concentration equation



(c) Total time

**Figure 5.16** – CPU time for different routines in Case 2.

## 5.4 EbFVM potential

Here, a simulation case with a more complex geometry is carried out. Reservoirs with complex geometry can be modeled with great flexibility with the proposed numerical technique. The hybrid unstructured grid used for discretizing this reservoir is shown in Fig. 5.17. In UTCHEM, this mesh cannot be represented using the original method. However, this geometry can be easily handled with the element-based formulation implemented in this work.

Local grid refinement in regions around the wells is considered for a more accurate solution in those locations. As stated before, this is one of the main advantages of using unstructured grids, because small elements can be concentrated only in localized areas of interest, without increasing excessively the size of the complete discrete problem. Moreover, with unstructured grids, the transition between refined and coarse regions can be performed smoothly, in order to avoid introducing further discretization errors associated with element sizes varying abruptly [28].

Two geological faults are present in the reservoir, and they are modeled as internal nearly impervious boundaries. The grid was enforced to conform to the domain boundary, as well as to the internal faults.
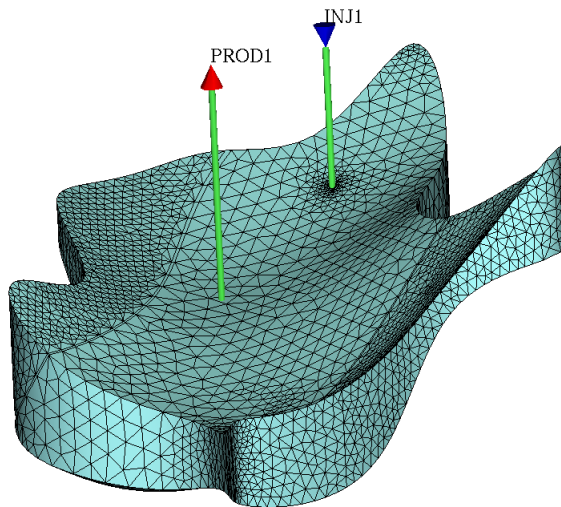


**Figure 5.17** – Grid.

In this example, a cylindrical region is employed around the wells, discretized with prisms and hexahedrons. In Flandrin *et al.* [21], this type of grid is suggested in order to capture the essentially radial flow pattern around the wells with more fidelity. The wells are represented by the lines in the middle of these regions, passing through the grid nodes. The transition between hexahedrons and tetrahedrons is performed using pyramids. Figure 5.18 shows where each element type is used in this grid discretization. Note that the faults are filled with hexahedrons.
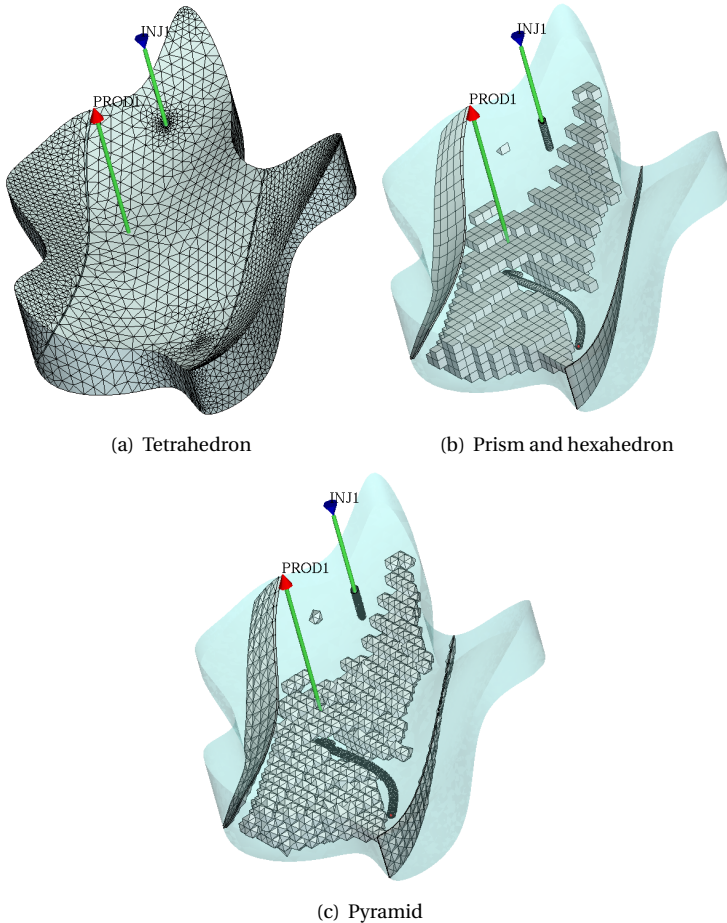


(a) Tetrahedron

(b) Prism and hexahedron



(c) Pyramid

**Figure 5.18** – Element types in the grid.

Figure 5.19 shows with more detail the region around the production well, where it is possible to further evidence the cylindrical region around it.



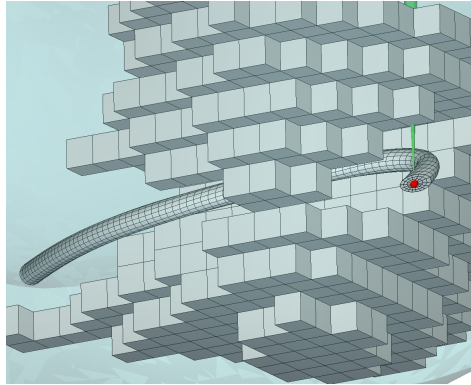**Figure 5.19** – Near-well region.

In order to visualize the flow behavior using the provided discretization, a water-oil flow problem is considered. The input data will not be shown, as the objective here is only to show qualitatively the fluid flow pattern. Figure 5.20 shows the saturation field after injecting 1.08 pore volumes of water.
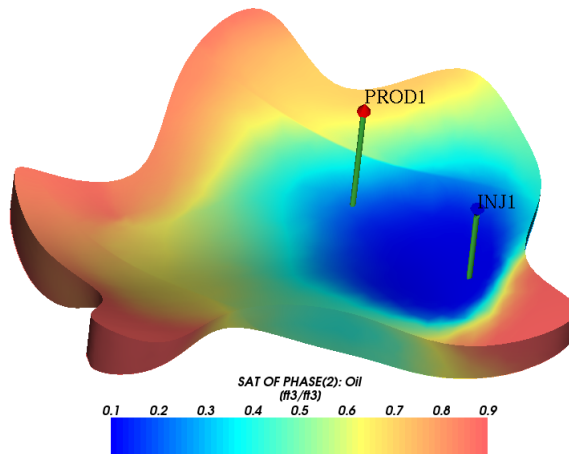


**Figure 5.20** – Saturation field in the final time (1.08 PVI).

Finally, Fig. 5.21 shows the saturation isosurfaces consisting of 50% each of water and oil at four different injection times. One can note that the simulation provides results that are qualitatively reproducing the expected physical behavior: the front of fluid is nearly radial around the injection well, and the displaced fluid is restrained by the faults, which have much lower absolute permeability values.



(a) 0.27 PVI                    (b) 0.54 PVI

(c) 0.81 PVI                    (d) 1.08 PVI

**Figure 5.21** – Isosurfaces of saturation 0.5.

# 6

# **CONCLUSIONS**

## 6.1 Summary

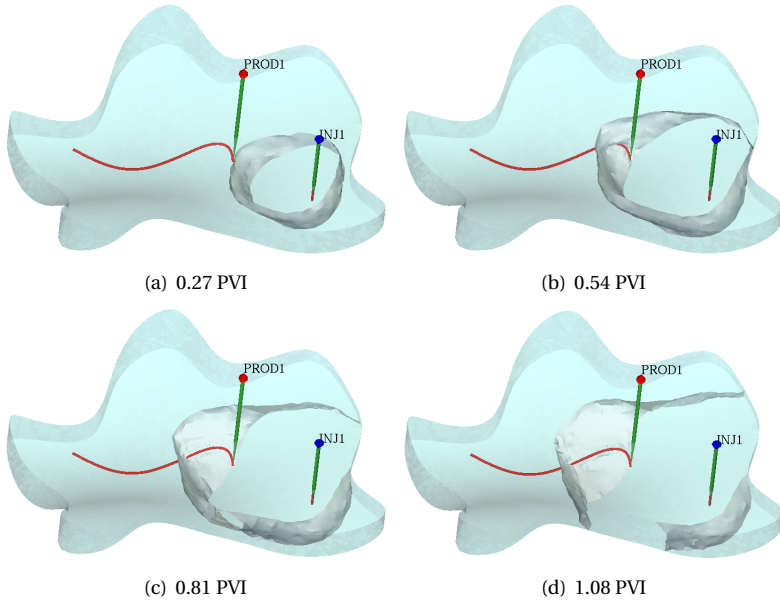This dissertation presented an application of the Element-based Finite Volume Method in a chemical flooding compositional simulator developed at The University of Texas at Austin. In this method, the discretization of the domain is realized in an unstructured manner using elements with different shapes, making it possible to represent complex geometries with high fidelity.

Since the main differences between the EbFVM and the FDM are related to geometrical matters, a solid base for the representation of geometrical entities considered in the formulation was established first. The discretization process for the PDEs was then realized following the philosophy of the conventional finite volume method, that is, the construction of discrete equations that satisfy the conservation of physical quantities at the control volumes. In order to take advantage of the element-based method, the integrated equations were written in a way they could be evaluated element by element, just by knowing the grid connectivity list. This way, the final equation for the control volumes can be assembled with contributions from the elements that are present in the control volume construction.

Originally in the simulator, the discretized equations were adapted to the use of a decoupled solution algorithm, where separate systems of equa-

tions are solved for the two primary variables, the aqueous phase pressure and the component concentrations. For this, the only scheme present in the simulator was the IMPEC algorithm, using the same timestep for both equations. In this method, the pressure is obtained implicitly, with explicit evaluation for the concentrations after solving for the pressure. This method is known as the fastest one on a pertimestep basis, however the stability criterion for the explicit evaluations always requires a limited timestep size. As this stability restrictions make the IMPEC at times inefficient or even impossible to solve some simulation problems, two other decoupled schemes were studied and implemented in UTCHEM. Only decoupled algorithms were implemented because the implementation of a fully implicit method, where the pressure and concentration are solved in a coupled way, would modify the whole code structure, not being worthy for the present work.

The first one is a variant of the IMPEC method, designated as A-IMPEC, where an acceleration strategy is proposed by taking larger timesteps for the pressure than for the concentrations, in order to solve the pressure linear system less times along the simulation. The method can be applied to any number of components, but only for incompressible problems. The other method considered is known as Sequential method, or Semi-Implicit method, where the concentration is also obtained implicitly, but in a separate linear system, after solving for the pressure implicitly in the same way as in the IMPEC. This method uses the same timestep for both equations, but allows for larger timesteps than the IMPEC, as the stability restriction was due only to the explicit evaluation of concentrations. This method does not seem to have any restriction; however, for the purpose of this work it was implemented only for water-oil flow problems.

## 6.2  Conclusions

The element-based conservative approach has shown to be efficient in all aspects of the objectives for the present work, evidencing its ability for solving petroleum reservoir problems in different configurations proposed in the examples. Although more computationally costly, the EbFVM presents several advantages in relation to traditional finite volume methods. The main advantage is the possibility of a loyal representation of complex geometries with intrinsic details, as present in real reser-

voirs, through the use of unstructured grids with local grid refinements. Furthermore, contrary to the original scheme present in the UTCHEM simulator, the method studied in this work applies multi-points in the flux calculation, which provides correct calculations in cases where the grid is non-orthogonal.

Regarding the timestepping approaches for accelerating the simulation, they demonstrated good performance. The necessity of a numerical formulation able to provide simulations in reasonable computational times motivated the consideration of such schemes. The A-IMPEC algorithm presents meaningful reductions in the time consumed for solving the pressure equation along the simulation. However, it does not provide any reduction in the time consumed for solving the concentrations. Consequently, in some cases, such as in chemical flooding, the time spent during the whole simulation is restricted by the concentration equation, which is larger than the time consumed by the pressure equation. When this happens, the total time-saving is not as significant as the time-saving in the pressure equation. The SEQ method, on the other hand, presented significant reductions for refined grids, being the most indicated, between the methods studied, for real reservoir problems.

The results obtained for both the application of EbFVM and the different timestepping approaches clearly show the capacity of such schemes in producing consistent solutions with several advantages compared to traditional schemes, which must motivate their use in other reservoir simulator applications. In the case of UTCHEM, the incorporation of these numerical schemes are an important addition to the simulator, which in terms of mathematical models already accounts for several petroleum engineering options.

## 6.3 Suggestions for future work

The following is a list of possible topics of study for future work in this area, or even in the UTCHEM simulator.

- Introduction of a full permeability tensor.

- Use of higher-order upwind interpolation schemes for the mobilities.

- Optimization of the implementation of the linear system assembling.

- A more rigorous analysis for the possibility of application of the A-IMPEC scheme in compressible problems.

- Extension of the SEQ method for problems with more than two components.

- Tests with other parameters for determination of the timestep in both the A-IMPEC and SEQ algorithms.

# REFERENCES

[1] AAVATSMARK, I.; KLAUSEN, R. A. **Well index in reservoir simulation for slanted and slightly curved wells in 3D grids**. *SPE Journal*, v. 8, n. 1, pp. 41–48, 1969.

[2] ABOU-KASSEM, J. H.; FAROUQ ALI, S. M.; ISLAM, M. R. **Petroleum reservoir simulation: a basic approach**. Gulf Publishing Company, Houston, 2006.

[3] AZIZ, K.; SETTARI, A. **Petroleum reservoir simulation**. Applied Science Publishers, London, 1979.

[4] BABU, D. K.; ODEH, A. S. **Productivity of a horizontal well**. *SPE Reservoir Engineering*, v. 4, n. 4, pp. 417–421, 1989.

[5] BALIGA, B. R.; PATANKAR, S. V. **A control volume finite-element method for two-dimensional fluid flow and heat transfer**. *Numerical Heat Transfer*, v. 6, pp. 245–261, 1983.

[6] BEAR, J. **Hydraulics of groundwater**. McGraw-Hill, New York, 1979.

[7] BHUYAN, D. **Development of an alkaline/surfactant/polymer flood compositional reservoir simulator**. Ph.D. Dissertation, The University of Texas at Austin, 1989.

[8] BLAIR, P. M.; WEINAUG, C. F. **Solution of two-phase flow problems using implicit difference equations**. *SPE Journal*, v. 9, n. 4, pp. 417–424, 1969.

[9] CAO, H.; AZIZ, K. **Performance of IMPSAT and IMPSAT-AIM models in compositional simulation**. *SPE paper 77720 – SPE Annual Technical Conference and Exhibition*, San Antonio, TX, 29 September – 2 October, 2002.

[10] CHEN, Z.; HUAN, G.; LI, B. **An improved IMPES method for two-phase flow in porous media**. *Transport in Porous Media*, v. 54, pp. 361–376, 2004.

[11] COATS, K. H. **Computer Simulation of Three-Phase Flow in Reservoirs**. The University of Texas at Austin, 1968.

[12] COATS, K. H. **Simulation of Steamflooding With Distillation and Solution Gas**. *SPE Journal*, v. 16, n. 5, pp. 235–247, 1976.

[13] COATS, K. H. **A note on IMPES and some IMPES based simulation models**. *SPE Journal*, v. 5, n. 3, pp. 245–251, 2000.

[14] COATS, K. H. **IMPES stability: selection of stable timesteps**. *SPE Journal*, v. 8, n. 2, pp. 181–187, 2003.

[15] CORDAZZO, J. **Simulação de reservatórios de petróleo utilizando o método EbFVM e multigrid algébrico**. Doctoral Thesis (in Portuguese), Federal University of Santa Catarina, 2006.

[16] CORDAZZO, J.; MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V. **The negative transmissibility issue when using CVFEM in petroleum reservoir simulation – 1. Theory**. *Proceedings of the 10th Brazilian Congress of Thermal Sciences and Engineering – ENCIT 2004*, Rio de Janeiro, 30 November – 3 December, 2004.

[17] CORDAZZO, J.; MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V. **The negative transmissibility issue when using CVFEM in petroleum reservoir simulation – 2. Results**. *Proceedings of the 10th Brazilian Congress of Thermal Sciences and Engineering – ENCIT 2004*, Rio de Janeiro, 30 November – 3 December, 2004.

[18] DELSHAD, M.; POPE, G. A.; SEPEHRNOORI, K. **A compositional simulator for modeling surfactant enhanced aquifer remediation**. *Journal of Contaminant Hydrology*, n. 23, pp. 303–327, 1996.

[19] DING, Y. **A generalized 3D well model for reservoir simulation**. *SPE Journal*, v. 1, n. 4, pp. 437–450, 1996.

[20] DOUGLASS, R. W.; CAREY, G. F.; WHITE, D. R.; HANSEN, G. A.; KALLINDERIS, Y. **Current views on grid generation: summaries of a panel discussion**. *Numerical Heat Transfer*, Part B, v. 41, pp. 211–237, 2002.

[21] FLANDRIN, N.; BOROUCHAKI, H.; BENNIS, C. **3D hybrid mesh generation for reservoir simulation**. *International Journal for Numerical Methods in Engineering*, v. 65, n. 10, pp. 1639–1672, 2006.

[22] FORSYTH, P. A. **A control-volume, finite-element method for local mesh refinement in thermal reservoir simulation**. *SPE Reservoir Engineering*, v. 5, n. 4, pp. 561–566, 1990.

[23] FOULSER, R. W. S.; GOODYEAR, S. G. **Improved stability of the IMPES formulation for chemical flooding**. *SPE paper 18417 – SPE Reservoir Simulation Symposium*, Houston, TX, 6–8 February, 1989.

[24] FUNG, L. S.; HIEBERT, A. D.; NGHIEM, L. **Reservoir simulation with a control-volume finite-element method**. *SPE Reservoir Engineering*, v. 7, n. 3, pp. 349–357, 1992.

[25] GOTTARDI, G.; DALL'OLIO, D. **A control-volume finite-element model for simulating oil-water reservoirs**. *Journal of Petroleum Science and Engineering*, v. 8, pp. 29–41, 1992.

[26] HURTADO, F. S. V. **Uma formulação de volumes finitos baseada em elementos para a simulação do deslocamento bifásico imiscível em meios porosos**. Master's Dissertation (in Portuguese), Federal University of Santa Catarina, 2005.

[27] HURTADO, F. S. V. **Formulação tridimensional de volumes finitos para simulação de reservatórios de petróleo com malhas não-estruturadas híbridas**. Doctoral Thesis (in Portuguese), Federal University of Santa Catarina, 2011.

[28] HURTADO, F. S. V.; MALISKA, C. R.; SILVA, A. F. C. **A variable timestep strategy for accelerating the IMPES solution algorithm in reservoir simulation**. *Proceedings of the XXVII Iberian Latin American Congress on Computational Methods in Engineering – CILAMCE 2006*, Belém, Brasil, 3–6 September, 2006.

[29] KARPINSKI, L.; MALISKA, C. R.; MARCONDES, F.; DELSHAD, M.; SEPEHRNOORI, K. **An element based conservative approach using unstructured grids in conjunction with a chemical flooding compositional reservoir simulator**. *Proceedings of COBEM 2009*, Gramado, 15–20 November, 2009.

[30] KARPINSKI, L.; TADA, M. P.; SILVA, A. F. C.; MALISKA, C. R.; SANSONI JR., U. **Peaceman's well model evaluation for non-isolated and partially penetrating wells**. *Rio Oil & Gas Expo and Conference 2010*, Rio de Janeiro, 13–16 September, 2010.

[31] KUZMIN, D.; MÖLLER, M. **Algebraic flux correction I. Scalar conservation laws**. In: Kuzmin, D.; Löhner, R.; Turek, S. (eds.). *Flux-corrected transport: Principles, algorithms, and applications*. Springer, Berlin, 2005.

[32] LAKE, L. W. **Enhanced oil recovery**. Prentice Hall, Englewood Cliffs, 1989.

[33] LABORATOIRE D'ANALYSE, TOPOLOGIE, PROBABILITÉS – MARSEILLE. **Benchmark on discretization schemes for anisotropic diffusion problems on general grids in 3D**. Available in http://www.latp.univ-mrs.fr/latp_numerique, accessed in November 2010.

[34] MACDONALD, R. C.; COATS, K. H. **Methods for Numerical Simulation of Water and Gas Coning**. *SPE Journal*, v. 10, n. 4, pp. 425–436, 1970.

[35] MAIZERET, P. D. **Well indices for non conventional wells**. Master's Thesis, University of Stanford, 1996.

[36] MALISKA, C. R. **Transferência de calor e mecânica dos fluidos computacional**, 2nd edition (in Portuguese). Livros Técnicos e Científicos Editora S.A., Rio de Janeiro, 2004.

[37] MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V.; DONATTI, C.; PESCADOR, A. V. B. **Projeto de Simulação de Reservatórios de Petróleo pelo Método EbFVM e Solver Multigrid (SimReP) – Rede de Gerenciamento e Simulação de Reservatórios (SiGeR)**. 1st Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2008.

[38] MALISKA, C. R.; SILVA, A. F. C.; TADA, M. P.; KARPINSKI, L.; PESCADOR, A. V. B.; JOSÉ, F. A.; SOPRANO, A. B.; VALE, B. T. **Desenvolvimento de um aplicativo para a simulação do escoamento acoplado poço-reservatório (WellRes) – Rede de Engenharia de Poços (REDEP)**. 2nd Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2009.

[39] MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V.; DONATTI, C.; PESCADOR, A. V. B. **Projeto de Simulação de Reservatórios de Petróleo pelo Método EbFVM e Solver Multigrid (SimReP) – Rede de Gerenciamento e Simulação de Reservatórios (SiGeR)**. 2nd Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2009.

[40] MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V.; DONATTI, C.; PESCADOR, A. V. B. **Projeto de Simulação de Reservatórios de Petróleo pelo Método EbFVM e Solver Multigrid (SimReP) – Rede de Gerenciamento e Simulação de Reservatórios (SiGeR)**. 3rd Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2009.

[41] MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V.; DONATTI, C.; PESCADOR, A. V. B. **Projeto de Simulação de Reservatórios de Petróleo pelo Método EbFVM e Solver Multigrid (SimReP) – Rede de Gerenciamento e Simulação de Reservatórios (SiGeR)**. 4th Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2009.

[42] MALISKA, C. R.; SILVA, A. F. C.; HURTADO, F. S. V.; DONATTI, C.; PESCADOR, A. V. B. **Projeto de Simulação de Reservatórios de Petróleo pelo Método EbFVM e Solver Multigrid (SimReP) – Rede de Gerenciamento e Simulação de Reservatórios (SiGeR)**. 5th Technical Report (in Portuguese), SINMEC Lab – Federal University of Santa Catarina, Florianópolis, 2009.

[43] OUYANG, L.-B.; AZIZ, K. **A simplified approach to couple wellbore flow and reservoir inflow for arbitrary well configurations**. *SPE paper 48936 – SPE Annual Technical Conference and Exhibition*, New Orleans, 27-30 September, 1998.

[44] PEACEMAN, D. W. **Fundamentals of numerical reservoir simulation**. Elsevier Scientific Publishing Company, Amsterdam, 1977.

[45] PEACEMAN, D. W. **Interpretation of well-block pressures in numerical reservoir simulation**. *SPE Journal*, v. 18, n. 3, pp. 183–194, 1977.

[46] PEACEMAN, D. W. **Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability**. *SPE Journal*, v. 23, n. 3, pp. 531–543, 1983.

[47] RAW, M. **A new control volume based finite element procedure for the numerical solution of the fluid flow and scalar transport equations**. Ph.D. Thesis, University of Waterloo, 1985.

[48] ROZON, B. J. **A generalized finite volume discretization method for reservoir simulation**. *SPE paper 18414 – SPE Reservoir Simulation Symposium*, Houston, TX, 6–8 February, 1989.

[49] SAAD, N. **Field scale studies with a 3-D chemical flooding simulator**. Ph.D. Dissertation, The University of Texas at Austin, 1989.

[50] SCHNEIDER, G.; ZEDAN, M. **Control-volume-based finite element formulation of the heat conduction equation**. *Spacecraft Thermal Control, Design and Operation*, v. 86, pp. 305–327, 1983.

[51] SORBIE, K. S. **Polymer-improved oil recovery**. CRC Press, Boca Raton, 1991.

[52] SPILLETTE, A. G.; HILLESTAD, J. G.; STONE, H. L. **A high-stability sequential solution approach to reservoir simulation**. *SPE paper 4542 - Fall Meeting of the Society of Petroleum Engineers of AIME*, Las Vegas, 30 September – 3 October, 1973.

[53] TADA, M. P. **Análise de esquemas numéricos de cinco e nove pontos para malhas tipo corner-point**. Master's Dissertation (in Portuguese), Federal University of Santa Catarina, 2009.

[54] TADA, M. P.; KARPINSKI, L.; SILVA, A. F. C.; MALISKA, C. R.; SANSONI JR., U. **Two dimensional off-center well modeling in reservoir simulation**. *Rio Oil & Gas Expo and Conference 2010*, Rio de Janeiro, 13–16 September, 2010.

[55] UTCHEM-9.0. **Technical documentation for UTCHEM-9.0: a three-dimensional chemical flood simulator**. Reservoir Engineering Research Program, Center for Petroleum and Geosystems Engineering. The University of Texas at Austin, 2000.

[56] UTCHEM-9.9. **User's guide for UTCHEM-9.9: a three-dimensional chemical flood simulator**. Reservoir Engineering Research Program, Center for Petroleum and Geosystems Engineering. The University of Texas at Austin, 2008.

[57] WOLFSTEINER, C.; DURLOFSKY, L. J.; AZIZ, K. **Calculation of well index for nonconventional wells on arbitrary grids**. *Computational Geosciences*, v. 7, pp. 61–82, 2003.

[58] ZGAINSKI, F.-X.; COULOMB, J.-L.; MARÉCHAL, Y.; CLAEYSSEN, F.; BRUNOTTE, X. **A new family of finite elements: the pyramidal elements**. *IEEE Transactions on Magnetics*, v. 32, n. 3, pp. 1393–1396, 1996 (in Portuguese).

# A

# PEACEMAN'S WELL MODEL FOR UNSTRUCTURED GRIDS

One of the pioneers in modeling the well in reservoir simulation accurately is Peaceman, who derived a relation between the well pressure and wellblock[1] pressure for uniform structured grids in 1978 [45]. As in most analytical techniques for well models, in order to determine the well index, a single-phase flow problem in a homogeneous and isotropic medium, with permeability $K$, is considered. Peaceman's model [45] also considers a steady-state regime. Therefore, for an areal reservoir model, with thickness $h$, in the well neighborhood the pressure will depend only on the radial coordinate $r$, as illustrated in Fig. A.1. Darcy's law, for these conditions, is reduced to

$$-\frac{Q}{2\pi rh} = -\frac{K}{\mu}\frac{dP}{dr} \ . \tag{A.1}$$

Integrating Eq. (A.1) between the well radius $r_w$ and a generic radius near the well, the following relationship for the pressure is obtained:

---

[1] As in Peaceman's work structured Cartesian grids are used, the well is located in the center of the gridblocks. A gridblock containing a well is denominated wellblock. In this dissertation, the well is located in the nodes, and, therefore, a node containing a well will be called wellnode.
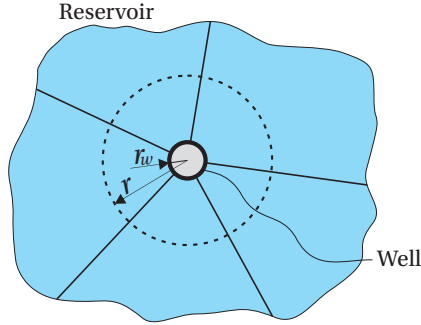
Reservoir



$r_w$

$r$

Well

**Figure A.1** – Well in an areal reservoir model.

$$P(r) = P_w + \frac{Q\mu}{2\pi hK} \ln\left(\frac{r}{r_w}\right) . \tag{A.2}$$

As performed in Peaceman [45], an equivalent radius, $r_o$, is associated to the wellnode. This is the radius at which the steady-state flowing pressure for the actual well is equal to the numerically calculate pressure for the wellnode, $P_p$. This definition for $r_o$ gives

$$Q = \frac{1}{\mu}\frac{2\pi hK}{\ln(r_o/r_w)}\left(P_p - P_w\right) . \tag{A.3}$$

The constant portion related to geometry and medium is grouped in a coefficient called well index, $WI$. Therefore, Eq. (A.3) can be written as

$$Q = \lambda WI\left(P_p - P_w\right) , \tag{A.4}$$

where $\lambda = \mu^{-1}$.

The determination of the well index is reduced, therefore, to the determination of the equivalent radius, location where the nodal pressure and the pressure from the analytical solution are equal. This determination for unstructured grids follows one of the ideas presented in Peaceman [45] for structured and uniform grids. The idea is to apply a mass balance over the control volume where the well is located and equate the flux over all faces of the control surface with the well flow rate from the analytical solution, given in Eq. (A.4). For this, considering a situation as illustrated in Fig. A.2, the well flow rate produced at the well must cross the faces in the control surface. Thus, applying Eq. (3.24) for one phase with constant
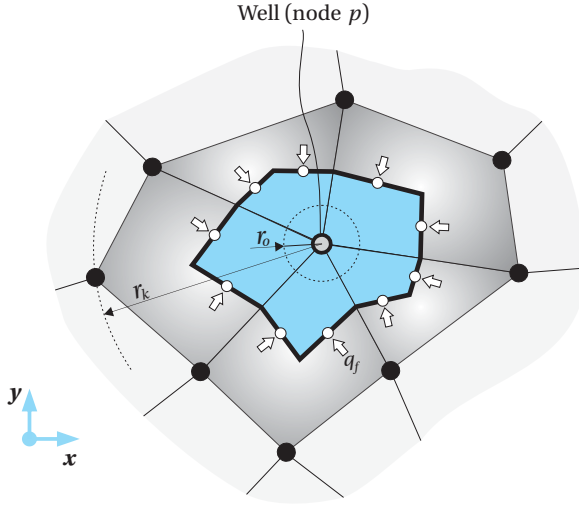
Well (node $p$)



**Figure A.2** – Control volume and elements around a producer well.

mobility in a 2D areal domain, one can obtain

$$Q = \sum_{f \in \mathbb{F}_p} q_f = \sum_{e \in \mathbb{E}_p} \sum_{f \in \mathbb{F}_p^e} \lambda [b]_{\hat{f}}^{T} [\hat{P}]_e \; . \tag{A.5}$$

In Eq. (A.5), $q_f$ are the individual fluxes through each of the faces that form the control surface. Since only one phase with constant viscosity is being considered, the subscripts related to the phase in Eq. (3.24) were omitted. After performing the summations in Eq. (A.5), a linear expression for the nodal pressures is obtained, which can be written in a similar way as performed in Eq. (3.28):

$$Q = \sum_{k \in \mathbb{T}_p} \lambda \, a_k P_k \; , \tag{A.6}$$

where $k \in \mathbb{T}_p$ represents all nodes in the stencil of node $p$ [40]. This stencil is formed by the nodes from the elements that contain node $p$, that is, from the elements that are present in the control volume construction. The coefficients $a_k$ are combinations of some components of vector $[b]_{\hat{f}}$. This group of coefficients has the important property that the summation over all coefficients is always null [37, 47], that is,

$$\sum_{k \in \mathbb{T}_p} a_k = 0 \; . \tag{A.7}$$

Subtracting from the RHS of Eq. (A.6), the term $P_p \sum a_k$, which is equal to zero, an equivalent relation is obtained:

$$Q = \sum_{k \in \mathbb{T}_p^*} \lambda \, a_k \left( P_k - P_p \right) \; , \tag{A.8}$$

where now the summation $k \in \mathbb{T}_p^*$ is only over the neighboring nodes. This equation, derived from the use of shape functions for the pressure gradient approximation, must be related to the analytical expression obtained by integrating Eq. (A.1). Assuming this equation is valid until the position where the wellnode neighbors are located, the integration of Eq. (A.1) from $r_o$ to $r_k$ will provide

$$P_k = P_p + \frac{Q\mu}{2\pi h K} \ln \left( \frac{r_k}{r_o} \right) \; , \tag{A.9}$$

where $P_k$ is the pressure of each node in the wellnode stencil and $r_k$ is the distance between node $k$ and the wellnode $p$. Substituting Eq. (A.9) into Eq. (A.8), one can obtain

$$Q = \sum_{k \in \mathbb{T}_p^*} \lambda \, a_k \frac{Q\mu}{2\pi h K} \ln \left( \frac{r_k}{r_w} \right) \; , \tag{A.10}$$

or

$$1 = \sum_{k \in \mathbb{T}_p^*} a_k \frac{1}{2\pi h K} \ln \left( \frac{r_k}{r_w} \right) \; . \tag{A.11}$$

Finally, from the previous expression it is possible to obtain the equivalent radius, which is given by

$$r_o = \exp \left( \frac{\displaystyle\sum_{k \in \mathbb{T}_p^*} a_k \ln r_k - 2\pi h K}{\displaystyle\sum_{k \in \mathbb{T}_p^*} a_k} \right) \; . \tag{A.12}$$

The present procedure provides the equivalent radius, and therefore, the well index for vertical wells in unstructured grids, which are represented by a point in 2D areal reservoirs. In order to apply such a

scheme for vertical wells in 3D unstructured grids, the reservoir must have only horizontal layers, in such a way the wells are perpendicular to the layers. This way, each layer is treated as a 2D areal reservoir, and the same procedure can be applied. Note that in this case, in order to perform the mass balance, the stencil of each wellnode is reduced to the neighbors located at the same layer. The nodes located on the top or bottom layers are ignored.