Pontifícia Universidade Católica
do Rio de Janeiro

Anderson Wang

Development of a compositional reservoir simulator based on a plugin architecture

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica.

Advisor: Prof. Ivan Fábio Mota de Menezes

Rio de Janeiro
June 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Anderson Wang

Development of a compositional reservoir simulator based on a plugin architecture

Dissertation presented to the Programa de Pós–graduação em Engenharia Mecânica of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Ivan Fábio Mota de Menezes
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Dr. Leonardo Seperuelo Duarte
Instituto Tecgraf de Desenvolvimento de Software
Técnico-Cientifico da PUC-Rio – Tecgraf/PUC-Rio

Dr. Daniel Nunes de Miranda Filho
Petróleo Brasileiro S.A. – Petrobras

Rio de Janeiro, June 06, 2024

Anderson Wang

Graduated in 2020 as a mechanical engineer from Pontifical University Catholic of Rio de Janeiro. While doing his Masters in Mechanical Engineering with an emphasis on Petroleum and Energy, he worked as a researcher at Tecgraf Institute/PUC-Rio developing simulators for the oil and gas industry.

Bibliographic data

Wang, Anderson

Development of a compositional reservoir simulator based on a plugin architecture / Anderson Wang; advisor: Ivan Fábio Mota de Menezes. – 2024.

104 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2024.

Inclui bibliografia

1. Engenharia Mecânica – Teses.
2. Simulador de reservatórios. 3. Modelo composicional. 4. Formulação de Coats. 5. Método das Diferenças Finitas. I. Menezes, Ivan Fábio Mota de. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

CDD: 621

To my beloved family for their support.

Acknowledgments

Words can hardly describe the gratitude to my advisors Ivan Menezes and Leonardo Duarte for their guidance, which I will carry with me for the rest of my professional career.

My deepest thanks go to Waldemar Celes Filho for giving me the opportunity to join his pleasant and enriching work group, where I was able to grow professionally.

I sincerely thank Petrobras, especially Daniel Nunes de Miranda Filho and Luiz Otávio Schmall dos Santos, who gave their precious time to help and make this research possible.

Special thanks to Eduardo Goicoechea and Renan Finotti for making GSIM's compositional version possible and for their wise help in revising my dissertation. Their support was essential in solving several issues during the development of the simulator.

I would like to thank the entire GERESIM team for sharing with me new experiences in software development.

Thanks for Thiago Bastos, Matheus Hoffmann and Henrique Santiago for their friendship. I will never forget our funny experiences together.

Finally, I would like to thank my parents, Andreia and Davi, and my sister, Angelica, for their love and care. Without them, I would not be who I am today.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Wang, Anderson; Menezes, Ivan Fábio Mota de (Advisor). **Development of a compositional reservoir simulator based on a plugin architecture**. Rio de Janeiro, 2024. 104p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

This work describes the development of a compositional simulator for petroleum reservoirs. It consists of a computational tool for numerical simulation of the dynamic behavior of reservoirs, aiming at increasing the efficiency of oil and gas extraction activities. The most well-known simulation formulation is the black oil, in which the migration of hydrocarbon components between different phases is captured by the solubility of one phase in another. However, there are cases where such modeling simplification does not allow for obtaining realistic results, especially in the presence of volatile oils that can easily present phase migrations or also when CO2 is injected into the reservoir. On the other hand, to solve this problem, several compositional simulation models were developed, where such migrations of hydrocarbon components were considered. In this context, the main objective of this work is to develop a computational system for simulating compositional oil reservoirs based on the Coats formulation. The implementation consists of an alternative to the black oil model, already existing in the GSIM simulator for reservoir simulation, developed in partnership between Petrobras and PUC-Rio. The proposed compositional model uses a discretization of the domain through orthogonal Cartesian meshes and the finite difference method to approximate the differential equations that govern the problem. Furthermore, aiming to increase the efficiency of simulations, GSIM was developed in the C++ language using a plugin architecture, which is optimized to allow faster and more efficient treatment of Tops associated with reservoir problems. Using the plugin system allows greater versatility in developing new functionalities. The proposed computational system will be tested in three different reservoir simulations, and the results will be compared with those obtained from the CMG-GEM commercial simulator.

Keywords

Reservoir simulator; Compositional model; Coats formulation; Finite Difference method; Plugin architecture.

Resumo

Wang, Anderson; Menezes, Ivan Fábio Mota de. **Desenvolvimento de um simulador composicional de reservatórios baseado em uma arquitetura de plugins**. Rio de Janeiro, 2024. 104p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho descreve o desenvolvimento de um simulador composicional de reservatórios de petróleo. Trata-se de uma ferramenta computacional para simulação numérica do comportamento dinâmico dos reservatórios, visando uma maior eficiência nas atividades de extração de óleo e gás. O modelo de simulação mais conhecido é o black oil, no qual se considera a migração de componentes de hidrocarbonetos entre diferentes fases através da solubilidadfe de um componente em outro. No entanto, em alguns casos, tal simplificação de modelagem não permite a obtenção de resultados realistas como, por exemplo, na presença de óleos voláteis, devido à facilidade com que os componentes podem migrar de fase, ou também quando é realizada uma injeção de CO2. Em contrapartida, visando solucionar este problema, diversos modelos de simulação composicional foram desenvolvidos, nos quais são consideradas tais migrações de componentes de hidrocarbonetos. Neste contexto, o objetivo deste trabalho é desenvolver um sistema computacional para simulação de reservatórios de petróleo baseado na formulação de Coats. A implementação consistirá em uma alternativa ao modelo black oil, já existente no simulador GSIM para simulação de reservatórios, fruto de uma parceria entre a Petrobras e a PUC-Rio. A modelagem composicional proposta utiliza uma discretização do domínio por meio de malhas Cartesianas ortogonais e o método das diferenças finitas na aproximação das equações diferenciais que governam o problema. Além disso, visando aumentar a eficiência das simulações, o GSIM foi desenvolvido em linguagem C++ utilizando uma arquitetura de plugins, a qual é otimizada para permitir um tratamento mais rápido e eficiente da Tops associada a problemas de reservatórios. A utilização do sistema de plugins permite uma maior versatilidade no desenvolvimento de novas funcionalidades. O sistema computacional proposto é testado em três diferentes simulações de reservatórios e os resultados são comparados com os obtidos por um simulador comercial, denominado CMG-GEM.

Palavras-chave

Simulador de reservatórios; Modelo composicional; Formulação de Coats; Método das Diferenças Finita.

Table of contents

List of figures

List of tables

List of Abbreviations

GSIM – GERESIM simulator

PDEs – Partial Differential Equations

IMPEC – Implicit Pressure and Explicit Composition

FD – Finite Differences

FV – Finite Volumes

TPFA – Two Point Flux Approximations

MPFA – Multi-Point Flux Approximations

FE – Finite Elements

MFD – Mimetic Finite Difference

FI – Fully Implicit

IMPES – Implicit in Pressure and Explicit in Saturations Scheme

AIM – Adaptative Implicit Method

SFI – Sequential Fully Implicit

GMRES – Generalized Minimal Residuals

AMG – Algebraic Multi-grid

BICGSTRAB – Bi-Conjugate Gradient Stabilized

CPR – Constrained Pressure Residual

DRS – Dynamic Row Sum

FVF – Formation Volume Factor

EOS – Equations of State

PVT – Pressure-Volume-Temperature

PPU – Phase-Potential Upwind

HU – Hybrid Upwind

FDM – Backward Finite Difference

IMPSAT – Implicit Pressure and Saturation and Explicit Component Mole

Fraction

BHP – Bottom Hole Pressure

NElem – Number of Elements

WI – Well Index

VLE – Vapor-Liquid Equilibrium

CSR – Compressed Sparse Row

List of Symbols

$C_1,\ C_2,\ CO_2$ – the pseudo-components

i – the hydrocarbon component

α – phases (i.e. water, oil and gas)

Ω – the continuity equations domain

$\mathbb{R}^3$ – the three-dimensional space

n_c – the number of components simulated

n_p – the number of phases simulated

ϕ – the rock porosity

ϕ^o – the rock porosity at the reference pressure

c_r – the rock compressibility

V_b – the element bulk volume

N_i – the number of moles of the component i per pore volume

N_w – the number of moles of the water component per pore volume

k_{r_α} – the relative permeability of the phase α

k_{r_w} – the relative permeability of the water phase

k_{r_o} – the relative permeability of the oil phase

k_{r_g} – the relative permeability of the gas phase

$k^o_{r_\alpha}$ – the end-point relative permeability of the phase α

k^o_{rc} – the relative permeability at the irreducible saturation of the oil phase

k_{row} – the intermediate and wetting system relative permeability

k_{rog} – the intermediate and nonwetting system relative permeability

μ_α – the viscosity of the phase α

μ_o – the viscosity of the oil phase

μ_g – the viscosity of the gas phase

μ_i – the viscosity of the component i

μ_α^* – the viscosity of the phase α at atmospheric pressure

ξ_α – the molar density of the phase α

ξ_w – the molar density of the water phase

P – pressure

P_α – the pressure of the phase α

P_w – the water pressure

p – the oil pressure

p_α – the pressure of the phase α

p^W – the well bottom hole pressure

p_b – the bubble point pressure

S_α – the saturation of the phase α

S_α^o – the residual saturation of the phase α

S_o – the oil saturation

S_g – the gas saturation

S_w – the water saturation

γ_α – the specific weight of the phase α

γ_w – the specific weight of the water phase

x_α^i – the mole fraction of the component i in hydrocarbon phase α

x_i – the oil molar fraction of the component i

y_i – the gas molar fraction of the component i

$\bar{\bar{k}}$ – the absolute permeability tensor of the control volume

q_i – the molar flow rate of the component i

q_w – the molar flow rate of the water component

D_α^i – the diffusive flux term

u_α – the volumetric phase velocity

Φ_α – the phase potential across the porous medium

ρ_α – the mass density of the phase α

g – the acceleration of gravity

Z – the depth of the fluid column

B_α – the formation volume factor of the phase α

R_{so} – the solution gas/oil ratio

Z_α – the compressibility factor of the phase α

R – the universal gas constant

T_{emp} – the reservoir temperature

ν_α – the molar volume of the phase α

ν_α^r – the reduced molar volume of the phase α

ν_l – the molar volume of the liquid phase

F_v – the gas-phase molar volume

n_α^i – the number of moles of the component i in the phase α

n_α – the total number of moles in the phase α

z_i – the overall molar fraction of the component i

ζ_α – the viscosity parameter of the phase α

ζ_i – the viscosity parameter of the component i

ρ_α^r – the reduced molar density of the phase α

M_{w_i} – the molecular weight of the component i

M_{w_α} – the molecular weight of the phase α

T_{s_i} – the reference temperature for the component viscosity computation

T_{c_i} – the critical temperature of the component i

T_{c_α} – the critical temperature of the phase α

P_{c_i} – the critical pressure of the component i

P_{c_α} – the critical pressure of the phase α

V_{c_i} – the critical volume of the component i

$\bar{\bar{k}}$ – tensor of the absolute permeability V_p – the pore volume

V_α – the volume of the phase α

t_α – the exponent of the relative permeability of the phase α

s, q, r, u, w – intermediate parameters for Z_α quantification

f_α^i – fugacity of the component i in the phase α

f_o^i – fugacity of the component i in the oil phase

f_g^i – fugacity of the component i in the gas phase

φ_α^i – the fugacity coefficient of the component i in a phase α

κ_{ij} – the binary interaction parameter between components i and j

a_i, b_i, f_w, a_α, b_α, A_α, B_α – intermediate parameters for φ_α^i quantification

g^* – the Gibbs' free energy

g_l^* – the Gibbs' free energy for liquid phase formation

g_v^* – the Gibbs' free energy for vapor phase formation

K_i – the equilibrium rate value of the component i

P_{r_i} – the reduced pressure of the component i

P_{c_i} – the critical pressure of the component i

T_{r_i} – the reduced temperature of the component i

T_{c_i} – the critical temperature of the component i

X_i – the mole numbers of component i in the oil phase

Y_i – the mole numbers of component i in the gas phase

A_o, A_g, f_{zi} – the intermediate parameters for a new phase appearance identification

R_o^i, R_g^i – the fugacity ratio correction for successive substitution update

ϵ – the tolerance of the stable thermodynamic equilibrium convergence

n_l – the number of moles of the liquid phase

n_v – the number of moles of the vapor phase

R_w – the well radius

G_f – the geometric factor

s – the skin factor

c – the circle factor

T – the transmissibility term

R_o – the equivalent well radius

J_ω – the productivity/injectivity index

T^g – the geometric transmissibility

λ_0 – the mobility factor

$l_1,\ l_2$ – the neighboring cells

$\overline{T}^g$ – the average geometric transmissibility

$\overline{\lambda}_{0,\alpha}$ – the average mobility transmissibility of the phase α

A – the normal area to the flow

Δt – the time interval

Δx – the control volume dimension along the x direction

$[\mathbf{J}]$ – the Jacobian matrix

$\mathbf{R}$ – the vector of residuals

1
Introduction

The oil and gas industry is one of the fundamental pillars of the global economy, playing a vital role in sustaining various industrial sectors and providing energy for modern society [1]. However, the oil industry faces constant challenges due to the complexity associated with the assessment and production from hydrocarbon reservoirs. These challenges involve optimizing the production and exploration processes while maintaining the environmental impact at a minimum level.

Hydrocarbon reservoirs are highly complex systems that contain a fluid mixture of different hydrocarbon components and water [2]. Given that reservoirs vary in geometry, composition, and thermodynamic properties, understanding the behavior of each unique reservoir is essential to optimize production, increase economic returns and minimize environmental impacts.

This way, reservoir simulators have emerged as indispensable tools to address the operational and strategic challenges faced by the petroleum industry, allowing engineers and geoscientists to predict the behavior of fluids in oil and gas reservoirs. The models behind those simulators present mathematical challenges associated with the increasing complexity in describing reservoirs as a more accurate representation is sought, as well as the often non-linear nature of the formulation adopted. Among these simulators, the ones based on the compositional reservoir model [3, 4] aim to improve the accuracy of previous model predictions, such as those based on the black oil formulation [5].

Despite the significant advances made in the development of compositional simulators in recent decades, the industry still faces several challenges that limit the effectiveness and applicability of these simulators. One of the challenges is the increasing complexity of reservoirs, including the presence of fluids such as heavy oils, extra heavy oils and sour gases, which exhibit complex thermodynamic and transport behavior. In addition, simulating large-scale reservoirs, such as those encountered in deep water and ultra-deep water, requires efficient and scalable computational solutions.

In this context, the main objective of this dissertation is to investigate and develop a versatile multi-purpose compositional version for the reservoir simulator called GSIM, covering its formulation, design and implementation

details. GSIM relies on a plugin-based architecture framework proposed by Duarte [6], called Topsim, specially designed for large-scale reservoir analysis written using the C++ programming language.

1.1 Literature Review

A reservoir simulator is a software that models fluid flow in an oil reservoir by approximating the solution of a set of continuous partial differential equations (PDEs) and algebraic equations. Those PDEs are obtained by combining the mass balance equation and Darcy's law, which is an empirically derived constitutive relation that describes laminar flow in porous media. However, the number of unknowns present in those PDEs can be greater than the number of differential equations available. To overcome this issue, geometric and thermodynamic considerations are used to define constraints, which are added in the form of the previously mentioned algebraic equations. Finally, the equations are solved using appropriate initial and boundary conditions.

Since computers cannot deal with continuous mathematical descriptions, discretization techniques are employed to approximate the solution to the governing fluid flow nonlinear PDEs in the simulator.

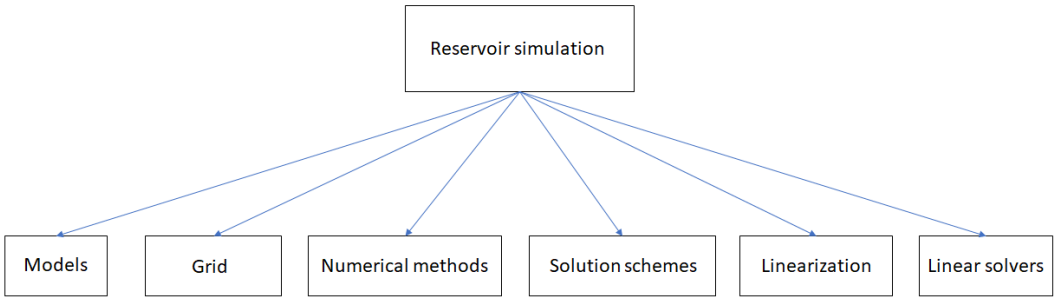

Figure 1.1: Elements involved in the solution workflow of a reservoir simulator.

1.1.1 Models

Although this dissertation is concerned with the development of a reservoir simulator that is based on a compositional model, for completeness, a brief history of the evolution of models and the different types found in the literature is presented.

The black oil formulation was one of the first reservoir models proposed, and it remains the most commonly used for simulations. It contains three

phases (liquid, vapor, and aqueous) and three pseudo components (oil, gas, and water). The model assumes isothermal flow, no volatility of the oil pseudo-component in the gas phase, and no mass exchange between the water phase and the other phases [5]. In other words, the water component is present only in the aqueous phase, and the oil component only in the oil phase. However, the gas component can exist in both gaseous and oil phases, undergoing dissolution and liberation processes [7].

Another formulation that was explored in Jacoby [8] is that of the volatile oil model. Volatile oils, also known as high-shrinkage crudes, are those found in natural conditions close to their critical temperatures. As a consequence, to accurately predict the amount of oil recovery, detailed knowledge of the separator conditions and the overall composition of the fluid entering the wellbore at each stage of depletion is required. For that reason, multi-component flash calculations are considered in that model. The results obtained are the amount and composition of the total wellstream fluid for a series of pressure increments, covering the production life of the reservoir. The total wellstream fluid is then flashed to separator conditions to obtain gas/oil ratio and volume of stock tank oil produced during each pressure increment. The model advantage is its ability to deal with intermediate fluids, between ordinary black oils and gases or gas-condensate fluids, which are found in deep high-temperature (250°F) reservoirs.

Nonisothermal models can also be found in the literature. In order to account for varying temperature, an energy conservation equation is also considered in the simulations. Therefore, the system of equations involve an additional temperature variable. Among the authors who treat this type of models, Delshad et al [9] discuss an efficient numerical method for multiphase nonisothermal flow using the iterative coupled Implicit Pressure and Explicit Composition (IMPEC) scheme. Its advantages over fully implicit nonisothermal compositional models are presented in Brantferger et al [10].

Finally, the compositional model is the one of interest in this dissertation. Unlike the black oil and volatile models, this model considers that fluid phases can be composed of any arbitrary number of pseudo-components. Therefore, it can provide a detailed representation of the behavior and composition of the reservoir fluids [2]. A complete discussion of this model is provided in Chapter 2.

1.1.2 Grid

As already mentioned, the continuous PDEs need to be discretized to obtain a numerical approximation to the solution. To do that, the PDEs' domain is partitioned into smaller regions, forming what is known as a grid. In addition, the derivatives in the PDEs are approximated by expressions that are evaluated at specific points of that grid (often at the centroids of each cell of the grid).

There are various ways to partition the reservoir domain and construct a grid. The most straightforward approach is to use a rectangular Cartesian grid with block-centered cells (i.e., the grid points are located at the center of the grid blocks).

Certain parts of the reservoir (e.g., around wells, regions characterized by high saturation variations) may require a higher level of refinement to maintain accuracy, justifying local grid refinements, as proposed in the literature [11–13].

For instance, it is common to adopt radial grids for representing regions close to wells. One of the reasons for this latter choice is that wells are dominated by radial flows [5, 14], and in that case, the flow is normal to the faces normal to the radial direction. Another reason is that the edges of a radial grid are easier to match with the edges of a circular well, whereas in a Cartesian grid following that geometry is more difficult and requires using smaller cells. This difficulty is illustrated in Fig. 1.2, where a graphical description of the errors added due to gridding is depicted.

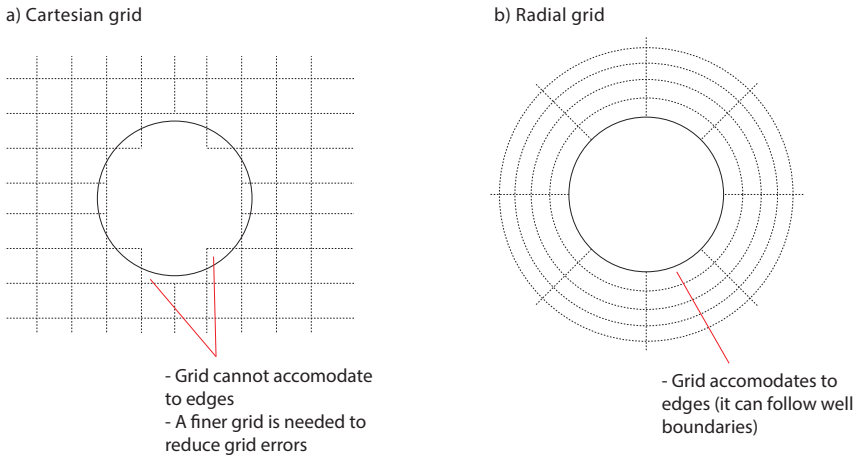

Figure 1.2: a) Planar view of a Cartesian grid around a well. The grid error is depicted. b) Planar view of a radial grid around a well. No grid error is added in this case.

Accompanying the boundaries in a complex reservoir geometry and the

presence of geological reservoir faults can justify adopting other types of grids, among which the corner-point representation is relevant. According to Aziz [15], the advantage of corner-point grids (over block centered grid, for example) is that the edges of the cells are free to take any arbitrary directions, and as a consequence, to match the direction of the complex boundaries or faults. This is similar to what was observed with radial grids and wells, and it represents an advantage over using Cartesian grids.

According to Farmer [16], although the several advantages provided by the corner-point flexibility, there are still some alignment problems related to overturned surfaces, intersecting faults and thrust faults unsolved. Aiming to solve these difficulties, recent studies suggest unstructured grids, which are based on a construction around a set of solution points that have no particular indexing scheme, as an alternative to the regular structured grid types described above [17, 18]. Figure 1.3 shows each type of grid discussed.

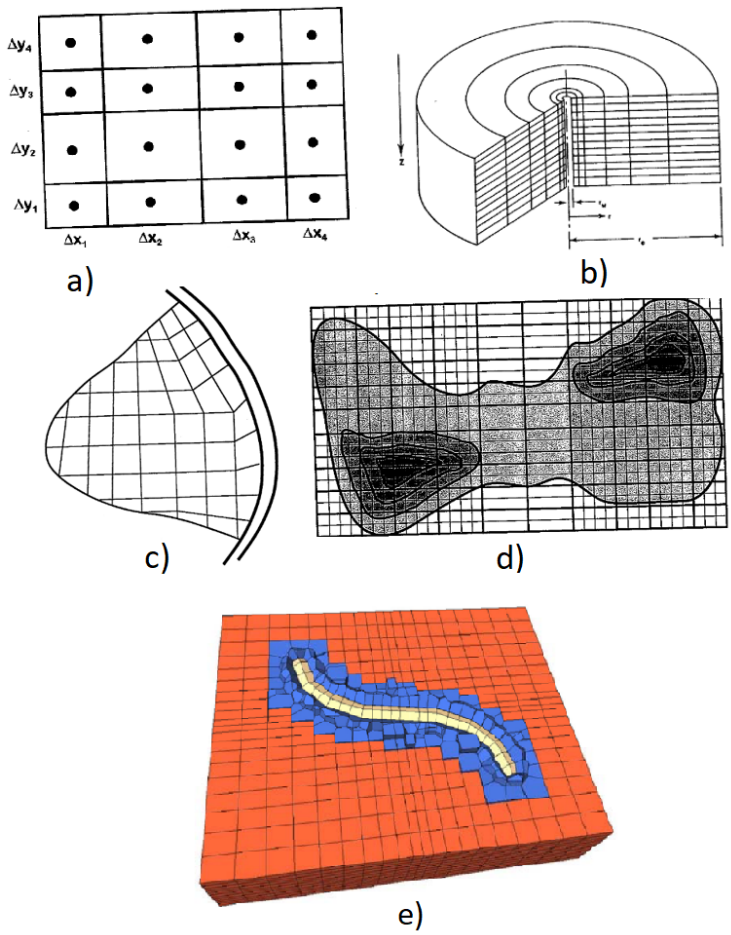

Figure 1.3: Different types of grid: a)rectangular Cartesian; b)radial; c)corner-point; d)hybrid local refinement; e) unstructured Voronoi grid embedded into a regular grid. The first four subcases are taken from Ertekin [5], whereas the last is taken from Flandrin et al. [19]

1.1.3 Numerical methods

The PDEs discussed earlier are highly nonlinear equations. Often, these equations cannot be solved using analytical techniques [20]. For that reason, numerical methods are used to discretize the PDEs and to obtain an approximate solution.

Discretization is needed for the time derivatives present in equations, as well as for the spatial derivatives involved. It is common to employ Finite Differences (FD) for the time derivatives [5, 21], leading to Backward or Forward Euler schemes. However, many possible alternatives are available to treat the spatial derivatives. For instance, the equations can be discretized directly from their weak form using FD, as well as from an integral mass balance formulation using the Finite Volumes (FV) method [22]. In the FV methodology, the grid elements are treated using control volumes, ensuring the mass balance is locally enforced [21]. When that path is followed, the derivatives involved are approximated using Two Point Flux Approximations (TPFA) [14], and more recently, Multi-Point Flux approximations (MPFA) [23]. Other possibilities are to apply the Finite Elements (FE) methods to the integral form of the mass balance, or Mimetic Finite Difference (MFD) [24].

1.1.4 Solution schemes

The choice of a backward or forward time integration leads to different solution schemes. Backward Euler integration is implicit: it generally leads to nonlinear equations in terms of unknowns evaluated at a future time step. On the contrary, forward integration is explicit, providing direct expressions to calculate the unknowns at a future time step in terms of past values [5].

Each integration scheme has advantages and disadvantages, particularly those related to the stability of the method. The higher the level of implicitness, i.e., taking a fully implicit (FI) method [25, 26], the more stable it is. While the lower the level of implicitness, as with a fully explicit method, the more computationally efficient it is to solve each time step. The drawback of the latter is that it requires using smaller steps to maintain solution stability.

A suitable choice of solution scheme for reservoir simulators is one that strikes a balance between stability and efficiency, giving rise to alternatives such as implicit in pressure and explicit in saturations scheme (IMPES) [27], adaptive-implicit method (AIM) [28–30], sequential fully implicit method (SFI) [31–37], among others. Section 2.3.3 addresses the presentation of

solution schemes and explain in detail those most commonly used to build compositional reservoir simulators.

1.1.5 Linearization methodology

After a solution scheme and a numerical method have been chosen to discretize the PDEs in time and space, a system of nonlinear equations is obtained. This nonlinearity is derived from the fact that specific terms present in these equations are also dependent of the unknowns. For instance, in compositional model, these terms are the phase transmissibility, capillary pressure, rock porosity, phase injection (production) rates, relative permeability, number of moles of component i per pore volume, hydrocarbon phase density and viscosity.

Therefore, before employs the linear solver methods (discussed in Subsection 1.1.6) to obtain the system solution, a linearization process is required. Considering

1.1.6 Linearization methodology

After a solution scheme and a numerical method have been chosen to discretize the nonlinear PDEs in time and space, a system of nonlinear equations is obtained. As a consequence, a methodology to find a solution of a nonlinear system of equations is required. The Newton-Raphson technique [21] can be used to overcome this difficulty. The aim of the method is to solve the nonlinear problem by taking iterative linear steps to approximate its solution.

Nevertheless, the solution of those intermediate linear approximations is a time-consuming and resource-intensive task as it requires the solution of a large, non-symmetric and sometimes ill-conditioned linearized set of equations. Typically between 60% to 80% of the total simulation time [29, 38–40] is used in this process.

The linear solvers used for those iterative steps of the Newton-Raphson method can be classified as either direct or iterative. The latter is considered almost mandatory for modern computer architecture, as they can deal with the limited scalability problem faced by direct solvers [21]. Some widely used iterative solvers in reservoir simulation are Orthomin [41], GMRES [42] and BICGSTRAB [43], which belong to the set of Krylov subspace algorithms.

In order to assure efficiency and robustness for the linear solver, the iterative solver is generally coupled with a preconditioner. It is a mathematical operator that accelerates the linear system solver convergence. Preconditioning

techniques can be divided into two subgroups: *local*, which is based on universally applicable algebraic methods such as Jacobi, Gauss-Seidel, Incomplete LU factorisation and Algebraic Multi-grid (AMG) [44]; and *global*, which is based on the analysis, geometry and physics of the problem, such as Constrained Pressure Residual (CPR) [45, 46] and Dynamic Row Sum (DRS) [47].

Figure 1.4 shows a complete scheme of the main steps for a reservoir simulation.

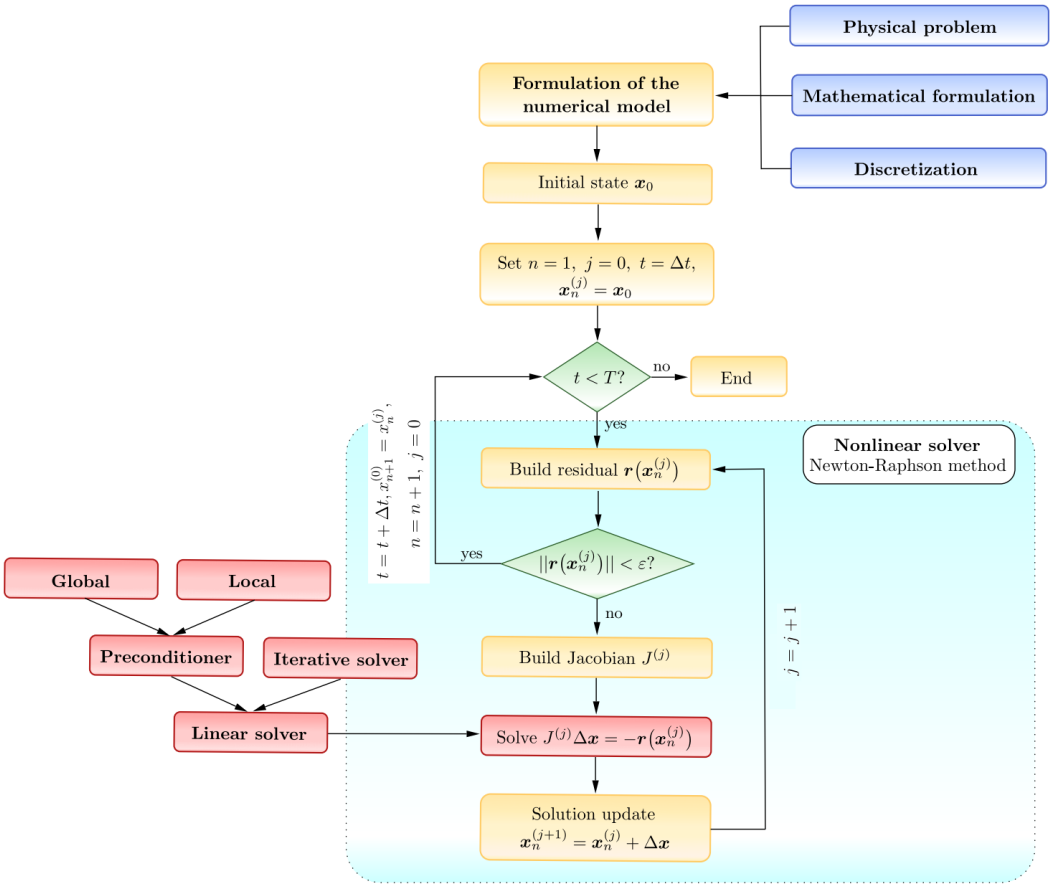

Figure 1.4: Main steps of a FI reservoir simulation, which t denotes time, T the end-point simulated time, Δt the time increment, ε the tolerance of the nonlinear solver and $\boldsymbol{x}_n^{(j)}$ is the vector formed by the problem unknowns with j the iteration counter and n the time step. Taken from [21].

1.2 Objectives and contribution

This dissertation continues the work of Bastos [48], which describes in detail the construction of the multi-purpose reservoir simulator called GERESIM Simulator (GSIM). It was based on the black oil model. The aim of this dissertation is to study and implement new functionalities in GSIM, adding

the capabilities of a compositional reservoir model and taking advantage of the characteristics of the plugin architecture based on the Topsim framework [6]. With the new implemented functionalities, GSIM can additionally:

- Switch between black oil or compositional models for the reservoir simulations. If the latter is selected, Coats' formulation is employed combined with an FI solution scheme;
- Switch between Corey's model [49] or Stone's model [50] for the relative permeability of the phase $k_{r\alpha}$ quantification;
- Switch between Soave-Redlich-Kwong [51] and Peng-Robinson(1976) [52] equations of state (EOS) for the Flash calculations.

Therefore, the main contribution of this work is to increase the range of functionalities that GSIM can simulate, providing flexibility while aiming to minimize major changes to existing codes by using a plugin architecture.

1.3 Thesis outline

This thesis is structured as follows. Chapter 2 presents the mathematical formulation of the compositional reservoir dynamics, detailing its differences from the black oil model. Chapter 3 describes the functionality of the plugin architecture using the Topsim framework integrated into GSIM reservoir simulator. In Chapter 4, the results of different reservoir simulations using the proposed implementation are presented. Also, the results are discussed and compared with those obtained with the CMG-GEM commercial simulator. Finally, Chapter 5 presents the conclusions of this research and suggestions for future work.

2
Mathematical model behind reservoir dynamics

Dynamic multiphase flow in a compositional reservoir simulation is considered fundamental for a complete understanding of the complex behavior of fluid systems, especially in subsurface porous medium. Therefore, this chapter covers the principles of fluid dynamics, thermodynamics and mathematical modeling to simulate the dynamic iteraction of multiple phases, such as liquid, vapor, and aqueous over time. An accurate representation of multiphase flow is essential for predicting fluid behavior, optimizing recovery strategies, and to assist in decision making processes associated with the exploration and production of reservoirs.

2.1
Multiphase flow

This dissertation focuses on the compositional reservoir model, which is considered as a general form of the black oil detailed by Bastos [48]. The main difference between them is that the former considers hydrocarbon phases as a combination of pseudo-components (e.g., C_1, C_2, CO_2, ...) within a multiflow stream, whereas the latter considers three pseudo-components: oil, present only in the liquid phase; water existing only in the aqueous phase; and gas that can be present in the liquid and vapor phases. This difference allows the compositional model to produce a more accurate prediction of the changes in composition over time and in different spatial regions of the reservoir. Also, the compositional model is best suited for simulations where the fluid composition is critical considerate, such as, gas-condensate reservoirs or advanced oil recovery situations. This way, the black oil model can be seen as a simplified and limited approach, while the compositional model is more comprehensive and accurate, covering a wider range of fluid conditions and behaviors. This concept is explained and emphasized in the following sections.

For better understanding, Figure 2.1 exemplifies the way both models treat a specific case of reservoir equilibrium. Notice that the compositional model is concerned with tracking and quantifying the amount of hydrocarbon components in a given phase.

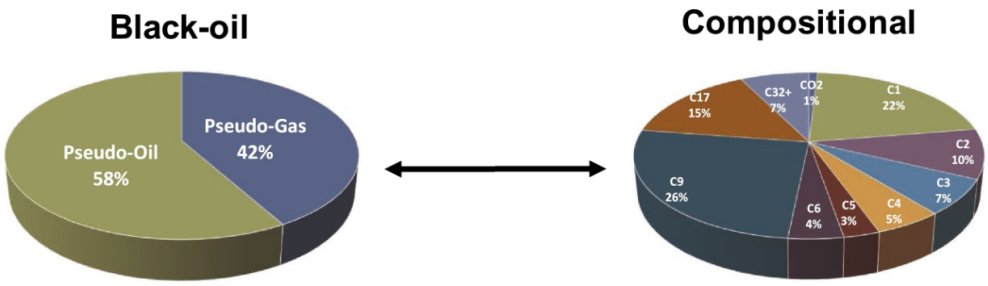

Figure 2.1: Different overviews between compositional and black oil models. Adapted from [7].

Chapter 4 will discuss selected test cases to be simulated in GSIM. It is important to mention that all simulations consider the effects of gravity. However, for this dissertation, the effect of capillary pressure and physical dispersion are not considered in the model. In the following sections, the letter i will be adopted to represent components, and α to represent the phases (i.e., liquid, vapor and aqueous).

2.1.1 Conservation of mass

To model fluid flow behavior in a porous medium, a continuity equation is formulated for each pseudo-component in the system. The reason for this is that the components can migrate between hydrocarbon phases, where in each new thermodynamic equilibrium, the phases can achieve different compositions. Hence, there is one material balance equation for each component i, given by:

$$V_b \frac{\partial}{\partial t}(\phi N_i) - V_b \vec{\nabla} \cdot \sum_{\alpha=2}^{n_p} \left(\frac{\bar{\bar{k}} k_{r_\alpha}}{\mu_\alpha} \xi_\alpha x_\alpha^i (\nabla P_\alpha - \gamma_\alpha \nabla D) - \underbrace{\phi \xi_\alpha S_\alpha \bar{\bar{K}}_{i\alpha} \nabla x_\alpha^i}_{D_\alpha^i} \right) - q_i = 0, \tag{2-1}$$

where ϕ is the rock porosity, N_i is the number of moles of i per pore volume and k_{r_α}, μ_α, ξ_α, P_α, S_α and γ_α are the relative permeability, viscosity, molar density, pressure, phase saturation and specific weight of phase α, respectively. In turn, x_α^i is the mole fraction of i in α, $\bar{\bar{k}}$ is the absolute permeability tensor of the control volume and q_i stands for the molar flow rate of i due to well injection

or production. Finally, following Fernandes [1], the diffusive flux term D_α^i will be neglected from Equation (2-1) due to the complexity of its measurement and implementation.

Here, water phase is treated as *immiscible*. This means that hydrocarbons can move freely between oil and gas phases for each new thermodynamic equilibrium, but not into the water phase. Therefore, there is no mass interchange between hydrocarbon and water phases [2]. The aqueous phase is interpreted as consisting only of a unique water component, with its own exclusive equation:

$$V_b \frac{\partial}{\partial t}(\phi N_w) - V_b \vec{\nabla} \cdot \frac{\bar{\bar{k}} k_{\mathrm{rw}}}{\mu_w} \xi_w (\nabla P_w - \gamma_w \nabla D) - q_w = 0, \tag{2-2}$$

where w denotes the water component.

The terms in Equations (2-1) and (2-2) physically represent, from right to left, the fluid accumulation, flow, and well, respectively. These terms arise from the combination of the continuity and Darcy's equations for the multiphase flow case [53]. The latter is expressed as:

$$\vec{u}_\alpha = -\bar{\bar{k}} \frac{k_{r\alpha}}{\mu_\alpha} (\nabla \Phi_\alpha), \tag{2-3}$$

where u_α is the volumetric phase velocity and Φ_α is the phase potential across the porous medium.

The general form for phase potential Φ_α in Equation (2-3) is:

$$\Phi_\alpha = P_\alpha - \rho_\alpha g Z, \tag{2-4}$$

where ρ_α is mass density of α, g is the acceleration of gravity and Z is the depth of the fluid column, with positive values downward.

2.1.2 Rock and Fluid Properties

The black oil and compositional models can often be used to analyze the dynamics of a specific reservoir from different perspectives. In this case, although the same reservoir is being analyzed, there are parameters/properties that are better suited for one model than the other. As an example, the properties of formation volume factor (FVF), B_α, and the solution gas/oil ratio R_{so}, once indispensable for black oil formulation, have a little or no application in compositional formulation. This occurs because the latter works with a detailed composition of hydrocarbon phases by components. In this case, as we would be dealing with hydrocarbons, concepts involving moles would be appropriate, such as, oil x_i and gas mole y_i fractions for each component.

In this way, it is worth noting that reservoir properties can be divided into fluid, rock, and rock/fluid properties, which are detailed in the following

subsections.

2.1.2.1 Fluid Properties

As mentioned above, the compositional model involves studying the dynamics of the hydrocarbon components in the reservoir fluids. In this sense, it would be necessary to use parameters that consider the influence of the components properties on the phases. The first parameter to be presented is the molar density of the phase, expressed as:

$$\xi_\alpha = \frac{P_\alpha}{Z_\alpha R T_{emp}}, \tag{2-5}$$

where Z_α is the compressibility factor, R is the universal gas constant, and T_{emp} reservoir temperature. In turn, ξ_α can also be defined as the inverse of molar volume ν_α.

The second parameter is the molar fraction x_α^i of i for each hydrocarbon phase α, expressed as the ratio between n_α^i and n_α. The former one is the number of moles of i in α, while the latter is the total number of moles in α. In this work, the molar fractions of oil and gas are referred as x_i and y_i, respectively. They are calculated as follows:

$$x_i = \frac{n_o^i}{n_o} \tag{2-6}$$

and

$$y_i = \frac{n_g^i}{n_g}. \tag{2-7}$$

Another necessary concept to introduce is the overall molar fraction z_i, given by:

$$z_i = \frac{\xi_o S_o x_i + \xi_g S_g y_i}{\xi_o S_o + \xi_g S_g}. \tag{2-8}$$

Equation (2-8) uses the concept of oil and gas saturation, S_o and S_g, respectively. This concept will be described in details in the Rock/Fluid Properties topic.

An important consequence from the molar fraction definition, is its constraint equation, expressed by:

$$\sum_{i=1}^{n_c} z_i = \sum_{i=1}^{n_c} x_i = \sum_{i=1}^{n_c} y_i = 1. \tag{2-9}$$

Then, the number of moles per pore volume, N_i, can be defined as:

$$N_i = \sum_{\alpha}^{n_p} \xi_\alpha S_\alpha x_\alpha^i, \tag{2-10}$$

where the sum operator works with α as liquid or vapor, skipping the aqueous phase, as shown by Santos [4].

Finally, several methodologies have been proposed to quantify the viscosity of hydrocarbon phases, μ_α. For example, Jossi *et al.* [54] suggest calculating μ_α by solving a fourth power equation. This suggestion considers that μ_α is a function of fluid composition, pressure and temperature. In the simulator proposed herein, two simplified definitions are used to calculate μ_α. The first is based on the sum of the product between each component viscosity μ_i and the corresponding phase molar fraction, i.e.:

$$\mu_o = \sum_{i=1}^{n_c} x_i \mu_i \tag{2-11}$$

and

$$\mu_g = \sum_{i=1}^{n_c} y_i \mu_i, \tag{2-12}$$

for oil and gas, respectively. The second is an empirical correlation presented in [55], given by:

$$\mu_\alpha = \mu_\alpha^* + \frac{1}{\zeta_\alpha}\left[\left(0.1023 + 0.023364\rho_\alpha^r + 0.058533(\rho_\alpha^r)^2 ... \right.\right.$$
$$\left.\left. - 0.40758(\rho_\alpha^r)^3 + 0.0093324(\rho_\alpha^r)^4\right)^4 - 0.0001\right], \tag{2-13}$$

where μ_α^*, ζ_α and ρ_α^r are the viscosity at atmospheric pressure, the viscosity parameter, and the reduced molar density of α, respectively. The term μ_α^* is obtained by the following equation:

$$\mu_\alpha^* = \frac{\sum_{i=1}^{n_c} z_i \mu_i^* \sqrt{M_{w_i}}}{\sum_{i=1}^{n_c} z_i \sqrt{M_{w_i}}}, \tag{2-14}$$

and the component viscosity at low pressure, μ_i^*, is given by:

$$\mu_i = \begin{cases} \frac{0.00034(T_{r_i})^{0.94}}{\zeta_i}, & \text{if } T_{s_i} \leq 1.5 \\ \frac{0.0001776(4.58T_{r_i} - 1.67)^{5/8}}{\zeta_i}, & \text{if } T_{s_i} > 1.5 \end{cases}. \tag{2-15}$$

The parameter T_{s_i} is calculated using the reservoir temperature T and the critical temperature of component T_{c_i}, i.e.:

$$T_{s_i} = \frac{T_{emp}}{\sum_{i=1}^{n_c} T_{c_i}}. \tag{2-16}$$

In turn, the viscosity parameter ζ_i of component i is calculated by:

$$\zeta_i = \frac{{T_{c_i}}^{1/6}}{M_{w_i}^{1/2} P_{c_i}^{2/3}}, \tag{2-17}$$

where M_{w_i} is the component molecular weight and P_{c_i} is the critical pressure

of i. To compute ζ_α, it is used an equation similar to (2-17), employing pseudoproperties specifically tailored for phases, as follows:

$$\zeta_\alpha = \frac{{T_{c_\alpha}}^{1/6}}{M_{w_\alpha}^{1/2} P_{c_\alpha}^{2/3}}. \tag{2-18}$$

These pseudoproperties T_{c_α}, M_{w_α}, and P_{c_α} are obtained as:

$$T_{c_\alpha} = \sum_{i=1}^{n_c} z_i T_{c_i}, \tag{2-19}$$

$$M_{w_\alpha} = \sum_{i=1}^{n_c} z_i M_{w_i}, \tag{2-20}$$

and

$$P_{c_\alpha} = \sum_{i=1}^{n_c} z_i P_{c_i}. \tag{2-21}$$

Finally, the reduced molar volume ν_α^r is calculated using the following equation:

$$\nu_\alpha^r = \nu_\alpha \frac{\sum_{i=1}^{n_c} z_i V_{c_i}}{M_{w_\alpha}}, \tag{2-22}$$

where V_{c_i} is the component critical volume.

2.1.2.2 Rock Properties

A main property of the rock is the porosity ϕ, interpreted as the ratio of pore volume to bulk volume in a rock sample. In a reservoir, ϕ can be divided into two categories: total porosity and effective porosity. While the former includes both isolated and interconnected pore spaces, the latter category includes only interconnected pore spaces [5]. In this sense, since only interconnected pores produce fluids, the ϕ property will be interpreted as the effective porosity category throughout this study. It is conventionally expressed as:

$$\phi = \phi^o \left[1 + c_r (P - P_o)\right], \tag{2-23}$$

where c_r is defined as the rock compressibility, P_o is the reference pressure, and ϕ^o is the porosity at P_o.

The second property to be introduced is the permeability. It is defined as the capacity of a porous medium to transmit its containing fluids through its interconnected pores. If this porous medium is completely filled with a single liquid phase, its capacity is known as the absolute permeability $\bar{\bar{k}}$, and due to be considered as a directional property, it can be expressed in the form of a tensor as follows:

$$\bar{\bar{k}} = \begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix}. \tag{2-24}$$

Rock/Fluid Properties

Another important concept to discuss is the fluid saturation S_α, described as the fraction of the pore space occupied by a given phase α. It can be expressed as the ratio of the fluid volume V_α to the pore volume V_p, given by:

$$S_\alpha = \frac{V_\alpha}{V_p}. \tag{2-25}$$

An important relation about the saturation property is that the sum of all saturations phases are equal to one. So, if a porous medium is completely filled with oil, gas and water phases, the following constraint emerges:

$$S_o + S_g + S_w = 1. \tag{2-26}$$

On the other hand, considering a porous medium filled by two or more fluids, it is necessary to introduce a new property called relative permeability $k_{r\alpha}$. Known as the fraction of the single phase that actually flows through the medium in a given direction, it is included in Equation (2-3). This inclusion is made to highlight the interference of a given fluid in the remaining fluid(s) [48].

Although there are various ways of quantifying the relative permeabilities of the phases, two methods are used in this work. The first is Corey's model for a three-phase system [49], given by:

$$k_{r\alpha} = k^o_{r\alpha} \left(\frac{S_\alpha - S^o_\alpha}{1 - \sum_{i=1}^{n_p} S^o_i} \right)^{t_\alpha}, \tag{2-27}$$

where $k^o_{r\alpha}$, S^o_α and t_α are the end-point relative permeability, the residual saturation, and the exponent of relative permeability of each phase α, respectively. In addition, the parameter n_p is the total number of phases simulated.

The second way would be to assume that $k_{r\alpha}$ are empirical functions depending on their respective saturations [56, 57], as:

$$k_{rw} = f(S_w) \tag{2-28}$$

and

$$k_{rg} = f(S_g). \tag{2-29}$$

Due to the difficulty to quantify the relative permeability of the intermediate phase (liquid) between the wetting (aqueous) and the non-wetting (vapor) phases, Stone proposed the three-phase model II [50] expressed by:

$$k_{ro} = k^o_{rc} \left[\left(\frac{k_{row}}{k^o_{rc}} + k_{rw} \right) \left(\frac{k_{rog}}{k^o_{rc}} + k_{rg} \right) - (k_{rw} + k_{rg}) \right]. \tag{2-30}$$

Finally, when the porous medium is saturated by two or more phases, the capillary pressure effect comes into account. This parameter is defined as the difference between the non-wetting fluid and the wetting fluid [58].

2.2 Phase Behavior

During the simulation, the simulator has to deal with the possibility that phases appear and disappear. In this respect, there are some differences in the detection methods between the black oil and compositional models. In the black oil, the main tool for detecting possible phase changes is the comparison between the oil pressure p and the bubble pressure p_b. However, in the compositional model, this analysis of phase behavior is more complex, which involves another aspects such as the equations of state, stability test and Flash calculation. These concepts will be described in the following subsections.

2.2.1 Cubic Equations of State

The compositional model uses cubic equations of state (EOS) to describe the thermodynamic behavior of multi-component systems. EOS are mathematical expressions that establish relationships between pressure, volume and temperature (PVT). These equations provide accurate descriptions of the volumetric and phase properties for both individual fluids and its combinations. They only require the critical properties and the acentric factor of each component w_i to describe the above mentioned properties [59].

There are various proposals for the EOS in the literature, such as, Soave-Redlich-Kwong [51] and Peng-Robinson (1976) [52]. Neverthless, their general structure uses the phase compressibility factor Z, given by:

$$Z^3 + sZ^2 + qZ + r = 0, \tag{2-31}$$

where:

$$\begin{cases} s = (u-1)\,B - 1 \\ q = A + (w-u)\,B^2 - uB \\ r = -AB - wB^2 - wB^3 \end{cases}. \tag{2-32}$$

The EOS models differ from each other by the values of u, w. This dissertation considers the Peng-Robinson (1976) EOS model to present all the parameters necessary to quantify the behavior of the hydrocarbon phases.

Equation (2-31) can be solved either by analytical methods or by trial and error approach, resulting in a maximum of three possible values for Z. Traditionally, the smallest root obtained (and greater than B) is typically

selected for liquids, while the largest root is preferred for vapors. The middle root is rejected due to its non-physical nature [59]. Regarding mixtures cases, the roots cannot be selected as mentioned earlier where the appropriated choice is made based on the root with the lowest normalised Gibbs' energy.

2.2.2 Fugacities

As seen in the previous sections, the hydrocarbon phases are formed by pseudocomponents for each stable thermodynamic equilibrium. This equilibrium condition, in turn, results from the minimization of the Gibbs free energy of the compositional system [60], which can be summarized as:

$$f_o^i = f_g^i, \tag{2-33}$$

where f_o^i and f_g^i are the fugacity functions of the component i in the oil and gas phases, respectively. In other words, f_α^i measures the leakage of component i from the hydrocarbon phase α to the other hydrocarbon phases. Therefore, in a stable thermodynamic equilibrium, the amount of component i forming an α phase would be equal to the amount of the same component leaving the other hydrocarbon phases to form the phase α.

The quantification of f_α^i is given by:

$$f_\alpha^i = p x_\alpha^i \varphi_\alpha^i, \tag{2-34}$$

where p is the oil pressure, φ_α^i is the fugacity coefficient of component i in the phase α, and x_α^i is a generic phase molar fraction of the component i.

Peng-Robinson EOS

In the Peng-Robinson EOS model, the parameters u and w, used in Equation (2-32), are 2 and -1, respectively. Once the Z_α values have been obtained using Equation (2-31), the φ_α^i can be calculated using:

$$\begin{aligned}\ln \varphi_\alpha^i = {} & \frac{b_i}{b_\alpha}(Z_\alpha - 1) - \ln(Z_\alpha - B_\alpha) \quad + \\ & - \frac{A_\alpha}{2\sqrt{2}B_\alpha}\left(\frac{2}{a_\alpha}\sum_{j=1}^{n_c} x_\alpha^j (1-\kappa_{ij})\sqrt{a_i a_j} - \frac{b_i}{b_\alpha}\right) \times \\ & \times \ln\left(\frac{Z_\alpha + (1+\sqrt{2})B_\alpha}{Z_\alpha - (1-\sqrt{2})B_\alpha}\right),\end{aligned} \tag{2-35}$$

$$\text{where} \quad \begin{cases} a_i = \frac{0.45724 R^2 T_{c_i}^2}{P_{c_i}} \left[1 + f_w \left(1 - \left(\frac{T_{emp}}{T_{c_i}} \right)^{0.5} \right)^2 \right] \\ b_i = \frac{0.07780 R T_{c_i}}{P_{c_i}} \\ f_w = 0.37464 + 1.54226\omega_i - 0.26992\omega_i^2 \\ a_\alpha = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} x_\alpha^i x_\alpha^j (1 - \kappa_{ij}) \sqrt{a_i a_j}, \qquad \alpha = o, g \\ b_\alpha = \sum_{i=1}^{n_c} x_\alpha^i b_i \\ A_\alpha = \frac{a_\alpha p}{R^2 T^2} \\ B_\alpha = \frac{b_\alpha p}{RT} \end{cases} . \tag{2-36}$$

Parameters a_i, b_i are empirical factors for component i, κ_{ij} is the binary interaction parameter between components i and j, R is the universal gas constant, and w_i is the acentric factor of component i [2].

2.2.3 Stability test

The compositional model considers hydrocarbon phases as a mixture of components. In this respect, the problem is to check when such a mixture would be ready to separate into more phases (i.e., oil and gas phases) at a given pressure and temperature. In this situation, the mixture is considered as unstable.

As a solution, the stability test is applied to all monophasic cells of the mesh in order to check which one contains a possible unstable mixture. Once the cell is identified, it is marked and redirected for the Flash procedure, which will quantify the new thermodynamic equilibrium compositions.

To achieve this purpose, the stability test proposes to use the Gibbs' tangent plane criterion to identify when the thermodynamic stability of a phase occurs. Figure 2.2 illustrates Gibbs' tangent plane criterion, which shows the behavior of the reduced Gibbs' energy g^* as a function of z_i. Considering the curve of the Gibbs' energy presented in Figure 2.2, a valid tangent plane cannot intersect Gibbs' energy surface anywhere except at the two points (i.e., vapor and liquid) of tangency (x_i, g_l^*) and (y_i, g_v^*). A physically valid two-phase solution is achieved when the composition of the mixture lies between the two equilibrium compositions x_i and y_i (as illustrated in Figure 2.2). If z_i (Figure 2.2) falls outside the range bounded by x_i and y_i (i.e., $z_i < x_i$ or $z_i > y_i$), it means that the present mixture is stable. Similarly, when the same criterion is achieved for mixtures, it also indicate stable conditions. If the overall composition z_i is between the equilibrium compositions (i.e., $x_i < z_i < y_i$), the mixture becomes unstable and separates into two equilibrium phases with compositions x_i and y_i [59, 61, 62].

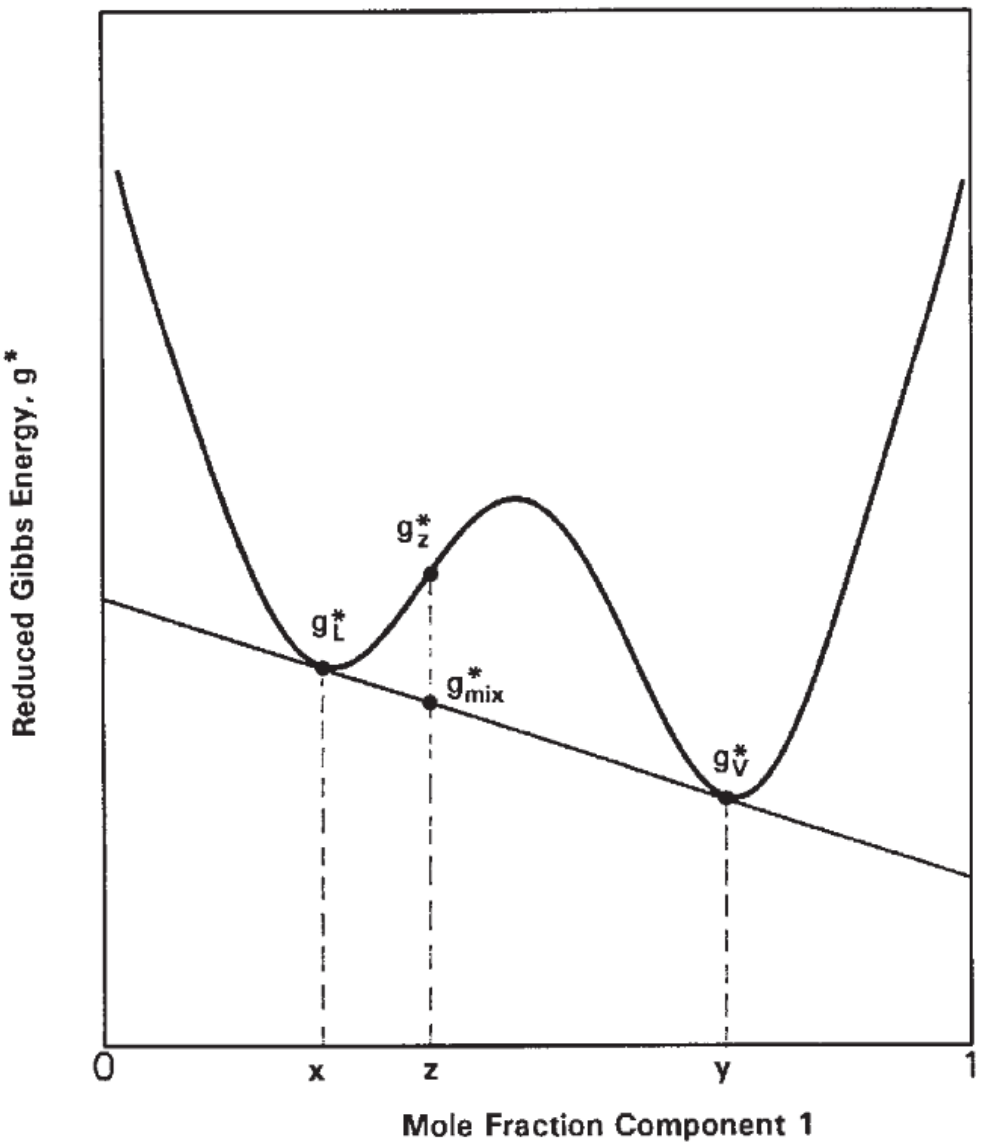

Figure 2.2: Gibbs' energy surface from a liquid-vapor system. Taken from Whitson [59].

The main steps for conducting the stability test are described below. It is important to emphasize that each test is performed separately to analyze one phase at a time, as stated in the work of Whitson & Brulé work [59].

First, it is necessary to calculate an estimate for the equilibrium rate values K_i for each component i.

In this sense, Wilson [63] suggests the following expression:

$$K_i = \frac{exp\left[5.37(1+\omega_i)\left(1-{T_{r_i}}^{-1}\right)\right]}{P_{r_i}}, \tag{2-37}$$

where P_{r_i} is the reduced pressure of component i, given by:

$$P_{r_i} = \frac{p}{P_{c_i}}, \tag{2-38}$$

and T_{r_i} is the reduced temperature as follows:

$$T_{r_i} = \frac{T_{emp}}{T_{c_i}}. \tag{2-39}$$

The next step is to calculate the second-phase mole numbers Y_i and sum them. Therefore:

$$\text{for oil phase,} \quad \begin{cases} X_i = z_i K_i \\ A_o = \sum_{i=1}^{n_c} X_i \end{cases} \quad \text{or for gas phase,} \quad \begin{cases} Y_i = z_i / K_i \\ A_g = \sum_{i=1}^{n_c} Y_i \end{cases}. \tag{2-40}$$

Then, after computing A_o and A_g parameters, it is necessary to compute the initial guess for x_i and y_i:

$$x_i = \frac{X_i}{A_o} \quad \text{or} \quad y_i = \frac{Y_i}{A_g}. \tag{2-41}$$

Further, the next step is to calculate the fugacity using z_i as a parameter (f_{zi}) and the second-phase fugacities (f_o^i or f_g^i) choosing their respective Z related to the lowest g^*. Once calculated, the fugacity ratio corrections (R_o^i or R_g^i) for successive substitution update are obtained by:

$$R_o^i = \frac{A_o f_o^i}{f_{z_i}} \quad \text{or} \quad R_g^i = \frac{f_{z_i}}{A_g f_g^i}. \tag{2-42}$$

If the convergence criteria[1] are met, as described below:

$$\sum_{i=1}^{n_c} (R_o^i - 1)^2 < 10^{-12} \quad \text{or} \quad \sum_{i=1}^{n_c} (R_g^i - 1)^2 < 10^{-12}, \tag{2-43}$$

the procedure stops, but if not, each K_i will be updated as:

$$K_i^{p+1} = K_i^p R_o^i \quad \text{or} \quad K_i^{p+1} = K_i^p R_g^i, \tag{2-44}$$

where p is the successive substitution update of K_i-value iteration counter.

The whole process described above refers to identifying the appearance of a new phase. In turn, the disappearance of a phase occurs when a saturation of a biphasic gridblock becomes negative. So, the negative saturation is set to zero and the gridblock becomes monophasic. In GSIM, when a gridblock is a biphasic type, for each Newton iteration, the saturation solutions are checked.

[1]The convergence criteria values are taken from Whitson [59].

2.2.4 Two-phase Flash calculation

As commented in the last subsection, the stability test identifies cells with a high chance of presenting unstable phases. A flash calculation is then performed on these potentially unstable cells to determine if phase splitting occurs and, in that case, to establish the composition of the phases at the new thermodynamic equilibrium. This Flash calculation involves applying the principles of EOS and fugacity, taking into account, for instance, the pressure p, temperature T, and the overall composition z_i of the system.

Michelsen [61] and Nghiem *et al.* [64] propose a methodology for performing the two-phase Flash calculation, satisfying the fugacity and material equilibrium constraints, using the Newton-Raphson algorithm. The aim is to reach the stable thermodynamic equilibrium, represented by Equation (2-33), by checking its convergence, expressed as:

$$\sum_{i=1}^{n_c} \left(\frac{f_o^i}{f_g^i} - 1 \right)^2 < \epsilon. \tag{2-45}$$

According to Whitson [59], its tolerance ϵ can be on the order of 10^{-13}.

The key steps in the two phase Flash calculation are shown below.

2.2.4.1 K-values estimation

The first step in the Flash calculation is to estimate the equilibrium rate K_i values for each component i. This calculation was already described earlier by Equation (2-37).

2.2.4.2 Rachford-Rice procedure

The aim of this step is to calculate the new values of x_i and y_i. For this purpose, it is necessary to introduce three relationships :

$$\begin{cases} z_i = \nu_l x_i + (1 - \nu_l) y_i; \\ x_i = \frac{z_i}{\nu_l + (1-\nu_l) K_i}; \\ y_i = K_i x_i, \end{cases} \tag{2-46}$$

where ν_l is the molar volume of the liquid phase (e.g., oil), obtained from the material balance constraint, given by:

$$\nu_l = \frac{n_l}{n_l + n_v}. \tag{2-47}$$

The parameters n_l and n_v represent the number of moles of liquid (e.g., oil) and vapor (e.g., gas), respectively. It is important to notice that for a set of three equations, there are three unknowns to find (ν_l, x_i and y_i). In this regard, the z_i value of each component i is given as an initial guess and recalculated after new values of x_i and y_i. By rearranging Equation (2-9), it yields:

$$\sum_{i=1}^{n_c} (x_i - y_i) = 0. \tag{2-48}$$

Applying the definition of z_i from Equation (2-46) to Equation (2-48), one arrives to the following expression:

$$F(\nu_l) = \sum_{i=1}^{n_c} \frac{(1 - K_i)z_i}{\nu_l + (1 - \nu_l)K_i} = 0. \tag{2-49}$$

Equation (2-49) is a monotonic function and its derivative $\partial F(\nu_l)/\partial \nu_l$ can be expressed analytically, so the Newton-Raphson algorithm is often used to solve for ν_l. This algorithm is expressed as:

$$\nu_l^{n+1} = \nu_l^n - \frac{F(\nu_l)}{F'(\nu_l)}, \tag{2-50}$$

where n is the Newton-Raphson's iteration counter for the Rachford-Rice procedure, and 0.5 is commonly used as an initial guess for ν_l.

Special care must be taken when implementing the algorithm. This is due to the fact that $F(\nu_l)$ has asymptotes along its function for each component i, which can lead to inconsistent results such as ∞^+ and ∞^- for ν_l, yielding a trivial solution. Figure 2.3 illustrates the problem above using the gas phase molar volume F_v instead of ν_l.

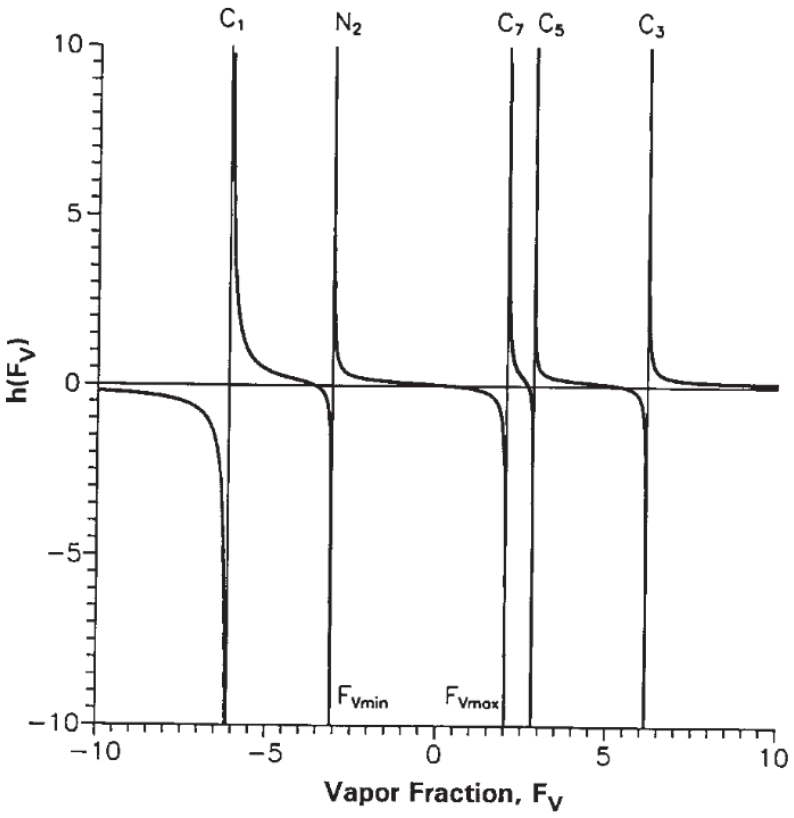

Figure 2.3: Rachford-Rice function $h(F_v)$ for a five-component mixture. Taken from Whitson [59].

For the convergence of ν_l, the following criterion[2] is used:

$$\left| \frac{\nu_l^{n+1}}{\nu_l^n} - 1 \right| < 10^{-8}. \tag{2-51}$$

Once ν_l is obtained, the parameters x_i and y_i can be obtained from Equation (2-46).

2.2.4.3 Fugacity parameters computation

This step is responsible for calculating all parameters related to EOS model implemented here and the fugacities. They were previously detailed in Sections 2.2.1 and 2.2.2.

2.2.4.4 Flash convergence

Once all the previous steps have been completed, the convergence must be checked for each component i using the criterion expressed in Equation

[2]The convergence criterion value is taken from [59].

(2-45). If the criterion is met, the Newton-Raphson iterative algorithm is stopped, otherwise new adjustments to the values of K_i must be made for each component i, i.e.:

$$K_i^{m+1} = K_i^m \frac{f_o^i}{f_g^i}, \tag{2-52}$$

where m is the Newton-Raphson iteration counter for Flash calculation. Once these adjustments have been made, all the above steps using the new values of K_i must be repeated until the convergence criteria is satisfied.

To summarize the Flash procedure, Figure 2.4 presents a diagram illustrating all the above mentioned steps.

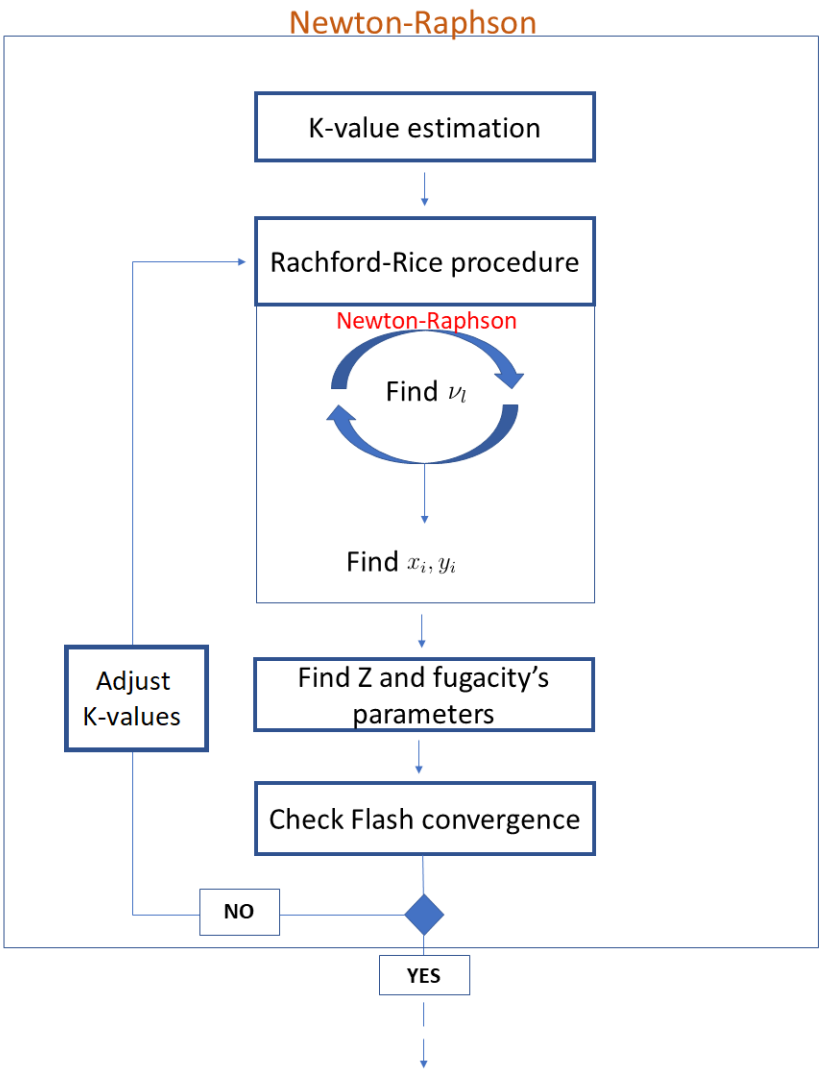

Figure 2.4: Main steps of a Flash routine.

2.3 Numerical solution

This section aims to discuss the numerical solution methods. In this regard, the mathematical model to be solved in the compositional model is

represented as:

$$\begin{cases} V_b \frac{\partial}{\partial t}(\phi N_i) - V_b \vec{\nabla} \cdot \sum_{\alpha=2}^{n_p} \left(\frac{\bar{\bar{k}} k_{r\alpha}}{\mu_\alpha} \xi_\alpha x_\alpha^i (\nabla P_\alpha - \gamma_\alpha \nabla D) \right) - q_i = 0 \\ V_b \frac{\partial}{\partial t}(\phi N_w) - V_b \vec{\nabla} \cdot \frac{\bar{\bar{k}} k_{\mathrm{rw}}}{\mu_w} \xi_w (\nabla P_w - \gamma_w \nabla D) - q_w = 0 \\ f_2^i - f_\alpha^i = 0, \quad \text{for } \alpha = 3...n_p \\ \sum_{i=1}^{n_c} x_\alpha^i = 1, \quad \text{for } \alpha = 2...n_p \\ \sum_{\alpha=1}^{n_p} S_\alpha = 1. \end{cases} \tag{2-53}$$

2.3.1 Treatment of interblock transmissibility and source/sink terms

In the context of fluid flow through porous media, transmissibility T is a crucial concept present in Equations (2-1)-(2-2). It is defined as the product of the geometric transmissibility, denoted as T^g, and the mobility factor λ_0. The geometric transmissibility is determined by the properties of the grid and the rock, remaining constant regardless of changes in fluid properties. On the other hand, the mobility factor, λ_0, depends on fluid properties. This relationship can be expressed as:

$$T = T^g \lambda_0. \tag{2-54}$$

The transmissibility terms, unlike the other properties shown in Section 2.1.2, are not exclusive properties of each cell, but of each interface between two neighboring cells l_1 and l_2. Figure 2.5 exemplifies this concept.

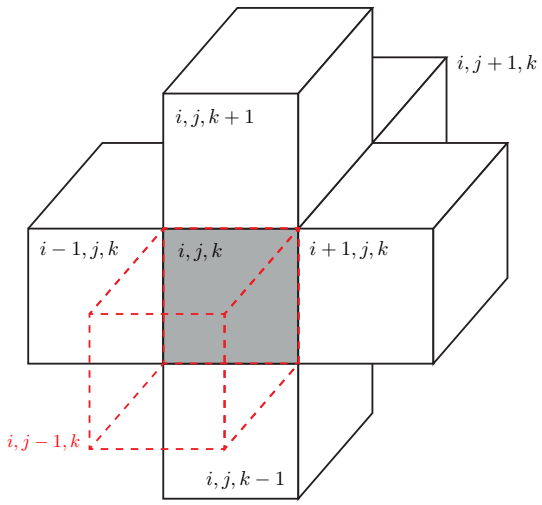

Figure 2.5: The transmissibility between cells (i, j, k) and (i, j - 1, k) is calculated in the demarcated gray region.

In this sense, it is necessary to apply averaging techniques for quantifying

T. Ertekin [5] describes in detail the calculation of T^g using harmonic averaging. To quantify the average λ_0, the upstream weighting technique is used, where the mobility of a phase at the interface between two gridblocks is equal to the mobility of the phase in the upstream cell.

The average geometric transmissibility between two neighboring l_1, l_2 $\overline{T}^g(l_1, l_2)$ and the average mobility $\overline{\lambda}_{0,\alpha}$ of a phase α at the interface between two cells are calculated as follows:

$$\overline{T}^g(l_1,l_2) = \frac{2}{\left(\frac{\Delta}{kA}\right)_{l_1} + \left(\frac{\Delta}{kA}\right)_{l_2}}, \quad \overline{\lambda}^{up}_{0,\alpha} = \begin{cases} \left(\xi_\alpha x^i_\alpha \frac{k_{r\alpha}}{\mu_\alpha}\right)_{l_1}, & \text{if } \Phi_{l+1} > \Phi_l \\ \left(\xi_\alpha x^i_\alpha \frac{k_{r\alpha}}{\mu_\alpha}\right)_{l_2}, & \text{otherwise} \end{cases}, \tag{2-55}$$

where $\overline{T}_{l+1/2} = \overline{T}(l+1, l)$, $\overline{T}_{l-1/2} = \overline{T}(l-1, l)$.

After the calculation of T, it is time to deal with the well term. Peaceman [65] suggests a simplified vertical well model as a numerical representation of the well term. In this sense, its equation is given as:

$$q_i = WI \cdot \sum_\alpha \left[\lambda_\alpha \xi_\alpha x^i_\alpha \left(p_\alpha - p^W\right)\right], \qquad \text{for} \quad \alpha = \text{oil, gas} \tag{2-56}$$

where WI is the constant well index, p_α is the pressure of the well block and p^W is the well bottom hole pressure for the well in the well block.

2.3.2 Discretization of the material balance equations

To solve the mathematical model presented, it is necessary to discretize the material balance equations (2-1)-(2-2). The initial step in constructing the numerical model involves integrating Equations (2-1)-(2-2), which can be expressed as follows [3, 4]:

$$\int_V V_b \frac{\partial}{\partial t}(\phi N_i)\, dV - \int_V V_b \vec{\nabla} \cdot \sum_{j=2}^{n_p} \left(\frac{\bar{\bar{k}} k_{rj}}{\mu_j} \xi_j x^i_j \left(\nabla P_j - \gamma_j \nabla D\right) - \phi \xi_j S_j \bar{\bar{K}}_{ij} \nabla x^i_j \right) dV... - \int_V q_i dV = 0 \tag{2-57}$$

and

$$\int_V V_b \frac{\partial}{\partial t}(\phi N_w)\, dV - \int_V V_b \vec{\nabla} \cdot \frac{\bar{\bar{k}} k_{rw}}{\mu_w} \xi_w \left(\nabla P_w - \gamma_w \nabla D\right) dV - \int_V q_w dV = 0. \tag{2-58}$$

Applying the Gauss theorem to the above equations, results in

$$\int_V \frac{\partial}{\partial t}(\phi N_i)\, dV - \int_A \sum_{j=2}^{n_p} \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i \bar{\bar{k}} \cdot (\nabla P_j - \gamma_j \nabla D) - \phi \xi_j S_j \bar{\bar{K}}_{ij} \nabla x_j^i \right) d\bar{A} ... - \int_V \frac{q_i}{V_b} dV = 0 \tag{2-59}$$

and

$$\int_V \frac{\partial}{\partial t}(\phi N_w)\, dV - \int_A \frac{k_{rw}}{\mu_w} \xi_w \bar{\bar{k}} \cdot (\nabla P_w - \gamma_w \nabla D)\, d\bar{A} - \int_V \frac{q_w}{V_b} dV = 0. \tag{2-60}$$

The Cartesian formulation employs hexaedral cell, assuming uniform properties within it. Consequently, the accumulation term in the preceding two equations can be integrated as follows:

$$\int_V \frac{\partial}{\partial t}(\phi N_i)\, dV = \left[\frac{\partial}{\partial t}(\phi N_i) \right]_{xyz} (\Delta x \Delta y \Delta z)_{xyz}\,, \tag{2-61}$$

$$\int_V \frac{\partial}{\partial t}(\phi N_w)\, dV = \left[\frac{\partial}{\partial t}(\phi N_w) \right]_{xyz} (\Delta x \Delta y \Delta z)_{xyz}\,. \tag{2-62}$$

Similarly, the source/sink term can be integrated as follows:

$$\int_V \frac{q_i}{V_b} dV = q_i \tag{2-63}$$

and

$$\int_V \frac{q_w}{V_b} dV = q_w, \tag{2-64}$$

where Peaceman's model, detailed in Section 2.3.1, gives the well model for nonsquare well blocks with anisotropic permeability.

Then, if the flux term in Equation (2-59) is expanded, it becomes:

$$\int_A \sum_{j=2}^{n_p} \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i \bar{\bar{k}} \cdot (\nabla P_j - \gamma_j \nabla D) \right) d\bar{A} = \sum_{j=2}^{n_p} \int_A \left[\frac{k_{rj}}{\mu_j} \xi_j x_j^i \bar{\bar{k}} \cdot (\nabla P_j - \gamma_j \nabla D) \right] d\bar{A}. \tag{2-65}$$

Assuming that the permeability tensor is diagonal, the integral of the term on the right-hand side of Equation (2-65) yields:

$$
\begin{aligned}
&\sum_{j=2}^{n_p} \int_A \left[\frac{k_{rj}}{\mu_j} \xi_j x_j^i \bar{\bar{k}} \cdot (\nabla P_j - \gamma_j \nabla D) \right] d\bar{A} \\
&= \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{xx} \right)_{x+\frac{1}{2}} \left(\frac{\partial P_j}{\partial x} - \gamma_j \frac{\partial D}{\partial x} \right)_{x+\frac{1}{2}} (\Delta y \Delta z)_{x+\frac{1}{2}} \quad + \\
&- \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{xx} \right)_{x-\frac{1}{2}} \left(\frac{\partial P_j}{\partial x} - \gamma_j \frac{\partial D}{\partial x} \right)_{x-\frac{1}{2}} (\Delta y \Delta z)_{x-\frac{1}{2}} \quad + \\
&+ \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{yy} \right)_{y+\frac{1}{2}} \left(\frac{\partial P_j}{\partial y} - \gamma_j \frac{\partial D}{\partial y} \right)_{y+\frac{1}{2}} (\Delta x \Delta z)_{y+\frac{1}{2}} \quad + \\
&- \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{yy} \right)_{y-\frac{1}{2}} \left(\frac{\partial P_j}{\partial y} - \gamma_j \frac{\partial D}{\partial y} \right)_{y-\frac{1}{2}} (\Delta x \Delta z)_{y-\frac{1}{2}} \quad + \\
&+ \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{zz} \right)_{z+\frac{1}{2}} \left(\frac{\partial P_j}{\partial z} - \gamma_j \frac{\partial D}{\partial z} \right)_{z+\frac{1}{2}} (\Delta x \Delta y)_{z+\frac{1}{2}} \quad + \\
&- \left(\frac{k_{rj}}{\mu_j} \xi_j x_j^i k_{zz} \right)_{z-\frac{1}{2}} \left(\frac{\partial P_j}{\partial z} - \gamma_j \frac{\partial D}{\partial z} \right)_{z-\frac{1}{2}} (\Delta x \Delta y)_{z-\frac{1}{2}} .
\end{aligned}
\tag{2-66}
$$

The final step in the development of the numerical model is the approximation of derivatives in Equations (2-61), (2-62) and (2-66). Considering the time derivative in Equations (2-61)-(2-62), the backward finite difference method (FDM) is used, while, the spatial derivatives in Equations (2-66) uses the central approximation of the FDM [5]. Therefore, the term inside the brackets in Equations (2-61) and (2-62) can be replaced by:

$$
\left[\frac{\partial}{\partial t} (\phi N_i) \right]_{xyz} \simeq \frac{1}{\Delta t} \left[(\phi N_i)^{k+1} - (\phi N_i)^k \right]_{xyz} , \tag{2-67}
$$

$$
\left[\frac{\partial}{\partial t} (\phi N_w) \right]_{xyz} \simeq \frac{1}{\Delta t} \left[(\phi N_w)^{k+1} - (\phi N_w)^k \right]_{xyz} , \tag{2-68}
$$

where the superscript k indicates the time step.

Furthermore, the spatial derivatives in the first term of the right-hand side of Equation (2-66) can be replaced by:

$$
\left(\frac{\partial P_j}{\partial x} \right)_{x+\frac{1}{2}} = \left(\frac{1}{\Delta x} \right)_{x+\frac{1}{2}} \left(P_{j_{x+1}}^{k+1} + P_{j_x}^{k+1} \right) , \tag{2-69}
$$

and

$$
\left(\frac{\partial D}{\partial x} \right)_{x+\frac{1}{2}} = \left(\frac{1}{\Delta x} \right)_{x+\frac{1}{2}} (D_{x+1} + D_x) . \tag{2-70}
$$

The derivatives for the other interfaces are approximated similarly.

Substituting Equations (2-69) and (2-70) in the first term on the right-hand side of Equation (2-66) gives:

$$\left(\frac{k_{rj}}{\mu_j}\xi_j x_j^i k_{xx}\right)_{x+\frac{1}{2}} \left(\frac{\partial P_j}{\partial x} - \gamma_j \frac{\partial D}{\partial x}\right)_{x+\frac{1}{2}} (\Delta y \Delta z)_{x+\frac{1}{2}} =$$
$$\left(\frac{k_{rj}}{\mu_j}\xi_j x_j^i\right)_{x+\frac{1}{2}}^{k+1} \left(\frac{\Delta y \Delta z}{\Delta x} k_{xx}\right)_{x+\frac{1}{2}} \left[\left(P_{j_{x+1}}^{k+1} - P_{j_x}^{k+1}\right) - \gamma_{j_{x+\frac{1}{2}}}^{k+1} \left(D_{x+1} - D_x\right)\right], \tag{2-71}$$

where $\left(\frac{k_{rj}}{\mu_j}\xi_j x_j^i\right)_{x+\frac{1}{2}}^{k+1}$ is calculated upwind and the relative permeability model is given by Corey's model.

Combining the equations for the accumulation, flux and well terms, gives the complete equation expressed as:

$$V\frac{\phi^{n+1}\sum\limits_{\alpha}\left(S_\alpha \xi_\alpha x_\alpha^i\right)^{n+1} - \phi^n \sum\limits_{\alpha}\left(S_\alpha \xi_\alpha x_\alpha^i\right)^n}{\Delta t}$$
$$= \left\{\sum_{s=1}^{ns}\left[T_s \sum_{\alpha}\left(\lambda_\alpha \xi_\alpha x_\alpha^i \Delta\Phi_\alpha\right)_s\right] + WI^W \cdot \sum_{\alpha}\lambda_\alpha \xi_\alpha x_\alpha^i \left[\left(p_\alpha - p^W\right)\right]\right\}^{n,n+1}, \tag{2-72}$$

which s and ns are a surface and a total number of surfaces.

Each term in both the flux and well terms can be evaluated at either time step n or time step $n+1$, depending on the implicitness of the resolution method selected.

2.3.3 Formulation methods

The number of variables to be solved in the nonlinear equations directly affects the degree of complexity of the problem, since it influences at the size of the Jacobian matrix, and consequently, increases the computational cost. Unlike the black oil model, which attempted to solve only three mass balance equations using the volume constraint equation, the compositional model solves $n_c(n_p-1)+n_p+1$ equations and variables for each biphasic gridblock. However, the extent to which dependencies can be explicitly eliminated and a condensed system of equations formulated may vary, depending on the complexity of the constraints. In cases where explicit elimination is not feasible, the variables to be solved must be classified as either primary or secondary variables. For the sake of clarification, some definitions are given below:

- **Primary variables:** A set of independent variables that contain exactly the minimum number of independent variables for the reservoir problem. Eventually, these are calculated using the primary equations;
- **Secondary variables:** Dependent variables that can be calculated once the primary variables are known. They are the remaining variables that serve as constraints that are essential for reducing the problem to the minimum set [3];

- **Primary equations:** Type of equations which, after linearization and combination with the linearized constraints through Gauss elimination, can be decoupled from the secondary ones, allowing the problem to be written in terms of the primary variables only;
- **Secondary equations:** The remaining equations used to find the secondary variables explicitly.

After having summarized the system of equations and their classification, it is necessary to revisit some solution methods, which deals with different degrees of implicitness. These include:

Fully-implicit method (FI): It is considered as an unconditionally stable method, which solves the primary variables and equations implicitly [3]. This method can accommodate significantly larger times teps than those used in explicit formulations. However, the larger the matrix derived from the fully implicit approach, the more computational resources per time step and core memory are required compared to explicit methods [26];

Implicit Pressure and Explicit Composition (IMPEC): This method is considered to be a much more stable alternative to the fully explicit method without significant computational cost. When the chosen primary variables are p and the composition, it solves p implicitly and the composition explicitly;

Implicit Pressure and Explicit Saturation (IMPES): Works with the same idea as IMPEC, but instead of explicitly solving the composition, it solves the saturation;

Implicit Pressure and Saturation and Explicit Component Mole Fraction (IMPSAT): Based on the idea of the coupling from gridblock to gridblock between the component mole fractions is weaker than the coupling between saturations. This method is presented as an upgrade of IMPES, where the pressure and saturations are simultaneously solved implicitly, and then the component mole fractions are updated explicitly gridblock by gridblock [66];

Adaptative Implicit Method (AIM): In order to reduce the computational cost, this method treats each gridblock differently, solving some with FI (in particular the gridblocks surrounding the wells) and the rest with IMPES.

Several formulations have been proposed to solve the set of equations (2-53). Each of them proposes a specific selection of variables and primary equations to be solved with different kind of implicitness degree. Accordingly to which they argue, their methodology can be the most efficient in terms of computational cost or for specific simulation situations. For example, Collins [67] chooses as primary variables p, N_i and N_w and as primary equations

the Equations (2-1), (2-2) and (2-26). Fernandes [1] summarized the different formulations in the literature, all of which are shown in Table 2.1.

	Implicitness degree	Primary variables
Fussel and Fussel (1979)	IMPEC	$\{P, \widetilde{N_g}, x_{2g}, ..., x_{n_c g}\}$ or $\{P, \widetilde{N_o}, x_{2o}, ..., x_{n_c o}\}$
Coats (1980)	FI	$\{P_g, S_o, S_g, x_{3g}, ..., x_{n_c g}\}$
Nghiem et al. (1981)	IMPEC	$\{P, \widetilde{N_w}, \widetilde{N_h}, z_1, ..., z_{n_c}\}$ or $\{P, S_w, \widetilde{N_h}, z_1, ..., z_{n_c}\}$
Young and Stephenson (1983)	IMPEC	$\{P, \widetilde{N_w}, \widetilde{N_h}, L_g, z_1, ..., z_{n_c-1}\}$
Chien et al. (1985)	FI	$\{P, \widetilde{N_w}, \widetilde{N_1}, ..., \widetilde{N_{nc}}\}$
Ács et al. (1985)	IMPEC	$\{P, \widetilde{N_w}, \widetilde{N_1}, ..., \widetilde{N_{nc}}\}$
Watts (1986)	IMPSAT	$\{P, S_o, S_g, \widetilde{N_w}, \widetilde{N_1}, ..., \widetilde{N_{nc}}\}$
Quandalle and Savary (1989)	IMPSAT	$\{P, S_w, S_g, x_{2o}, ..., x_{n_c-1o}\}$ or $\{P, S_w, S_g, x_{2g}, ..., x_{n_c-1g}\}$
Collins et al. (1992)	AIM/FI/IMPEC	$\{P, \widetilde{N_w}, \widetilde{N_1}, ..., \widetilde{N_{nc}}\}$
Branco and Rodriguez (1996)	IMPSAT	$\{P, S_w, S_g, x_{1o}, ..., x_{n_c-2o}\}$
Wang et al. (1997)	FI	$\{P, \widetilde{N_w}, \widetilde{N_1}, ..., \widetilde{N_{nc}}, \ln(K_1), ..., \ln(K_{nc})\}$
Haukas et al. (2007a)	IMPSAT	$\{P, S_w, S_g, x_1, ..., x_{n_c-n_p}\}$
Ayala (2004)	FI	$\{P, S_w, z_1, ..., z_{n_c-1}\}$

Table 2.1: Different types of formulation presented. Taken from Fernandes [1].

The simulator proposed herein uses the formulation presented by Coats [68], which will be described below.

2.3.3.1 Coats' formulation

To solve the set of Equations (2-53), Coats [68] proposes the selection of a set of primary variables using the FI solution method. Also known as natural variables, for biphasic gridblocks these are composed of p, S_o, S_g and $(n_c - 2)$ mole fractions of y_i, for $i = 3, ..., n_c$. Equations (2-1)-(2-2) were selected as primary equations, making the other remaining equations from the (2-53) set secondary [69]. Therefore, the full Jacobian matrix is computed according to the selected variables, followed by a Gauss elimination scheme to decouple the secondary variables (y_1, y_2 and x_i, for $i = 1, ..., n_c$) from the primary variables. After this elimination, the system is reduced to (n_c+1) equations per gridblock.

To perform the above operation, Coats checked that the set of equations (2-53) were functions depending on p, S_α, x_i and y_i. So if applying Equation (2-26) to eliminate one of the saturations, the diagonal submatrices of the entire Jacobian matrix (including primary and secondary variables) and the corresponding right-hand side for each gridblock can be written as shown in Figure 2.6.

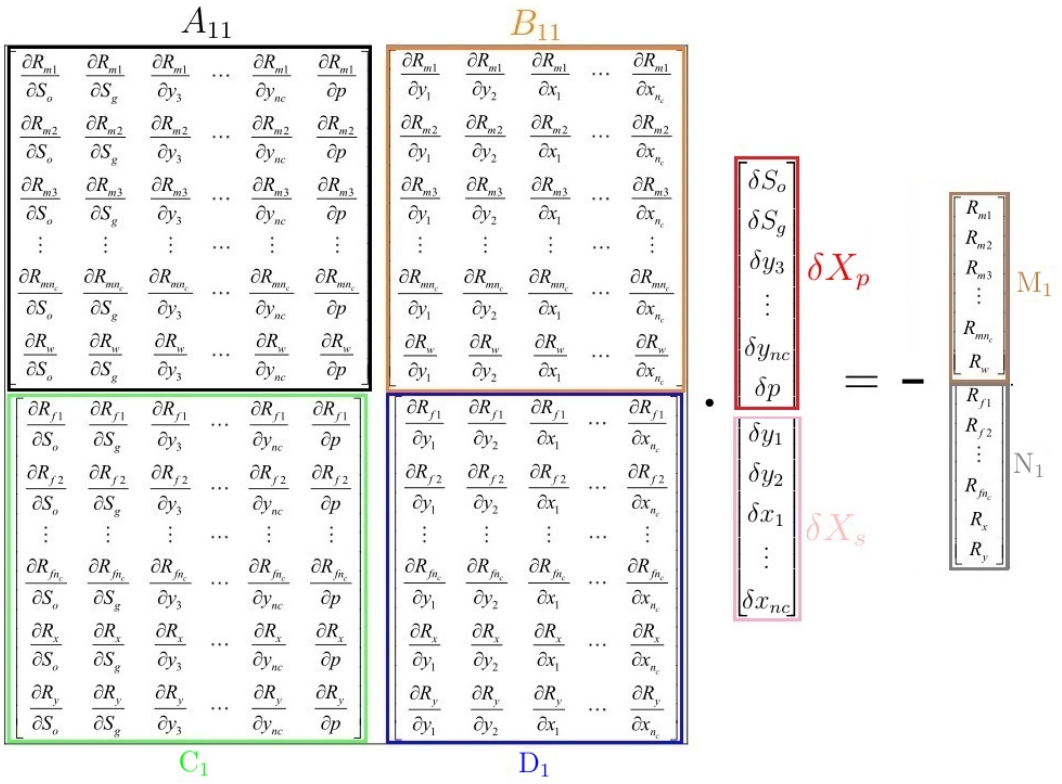

Figure 2.6: Diagonal submatrices of entire Jacobian matrix for biphasic gridblocks. Taken from Santos [4].

From Figure 2.6, it can be seen that each biphasic gridblock can be represented by four submatrices, which are: the derivative of the primary equations with respect to the primary variables ($\mathbf{A_{11}}$), the derivative of the primary equation with respect to the secondary variables ($\mathbf{B_{11}}$), the derivative of the secondary equations with respect to the primary variables ($\mathbf{C_1}$), and the derivative of the secondary equations with respect to the secondary variables ($\mathbf{D_1}$), as well as the residuals of the primary equations ($\mathbf{M_1}$) and the residuals of the secondary equations ($\mathbf{N_1}$).

The choice to name the first two submatrices as ($\mathbf{A_{11}}$) and ($\mathbf{B_{11}}$), and not as ($\mathbf{A_1}$) and ($\mathbf{B_1}$) respectively, concerns the flux term present in Equation (2-1) and discussed in Section 2.3.1. As seen, the flow part is not an exclusive characteristic of each gridblock, but of each connection between this element and each of its neighbors. In this case, it is also necessary to quantify the variation of the central element residual caused by the variation of the neighboring elements variables. For a better understanding, Figure 2.7 has been constructed.

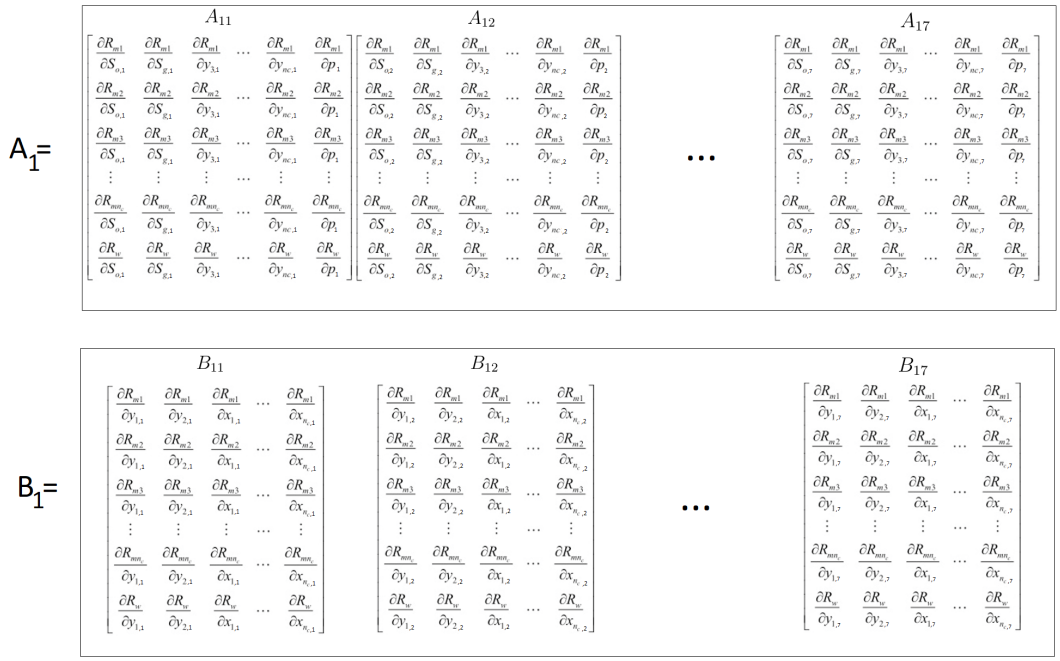

Figure 2.7: Submatrices of A and B of element 1 due to flux term computation.

Considering a mesh of elements in the classical 8-node hexahedron geometry, each gridblock can have up to six neighboring elements, which would add up to six additional submatrices (*off-diagonal submatrices*) to the main submatrix, responsible for storing the flux information. Figure 2.7 shows the maximum number of submatrices that can make up $\mathbf{A_1}$ and $\mathbf{B_1}$, using the following terminology: $\mathbf{A_{12}}$ indicates the submatrix whose residual equations of the main element (1) are perturbed by the primary variables of its first numbered neighboring element (2) (e.g. $\frac{R_{m1}}{S_{o,2}}$).

Rewriting the equation of Figure 2.6 as:

$$\begin{bmatrix} A_{11} & B_{11} \\ C_1 & D_1 \end{bmatrix} \begin{bmatrix} \delta X_p \\ \delta X_s \end{bmatrix} = - \begin{bmatrix} M_1 \\ N_1 \end{bmatrix}, \tag{2-73}$$

the Gauss elimination method could be used, in which the secondary variables can be eliminated using the procedure illustrated below:

$$\begin{bmatrix} A_{11} & B_{11} \\ D_1^{-1}C_1 & D_1^{-1}D_1 \end{bmatrix} \begin{bmatrix} \delta X_p \\ \delta X_s \end{bmatrix} = - \begin{bmatrix} M_1 \\ D_1^{-1}N_1 \end{bmatrix}, \tag{2-74}$$

$$\begin{bmatrix} A_{11} & B_{11} \\ B_{11}D_1^{-1}C_1 & B_{11} \end{bmatrix} \begin{bmatrix} \delta X_p \\ \delta X_s \end{bmatrix} = - \begin{bmatrix} M_1 \\ B_{11}D_1^{-1}N_1 \end{bmatrix}, \tag{2-75}$$

$$\begin{bmatrix} A_{11} - B_{11}D_1^{-1}C_1 & 0 \\ B_{11}D_1^{-1}C_1 & B_{11} \end{bmatrix} \begin{bmatrix} \delta X_p \\ \delta X_s \end{bmatrix} = - \begin{bmatrix} M_1 - B_{11}D_1^{-1}N_1 \\ B_{11}D_1^{-1}N_1 \end{bmatrix}, \tag{2-76}$$

$$\begin{bmatrix} A_{11} - B_{11}D_1^{-1}C_1 & 0 \\ D_1^{-1}C_1 & I \end{bmatrix} \begin{bmatrix} \delta X_p \\ \delta X_s \end{bmatrix} = - \begin{bmatrix} M_1 - B_{11}D_1^{-1}N_1 \\ D_1^{-1}N_1 \end{bmatrix}. \tag{2-77}$$

Using Equation (2-77), the primary equations are completely decoupled from the secondary variables, making it possible to solve them. Once the primary variables have been solved, the secondary variables of each gridblock are explicitly updated as follows:

$$\delta X_s = -D_1^{-1}N_1 - D_1^{-1}C_1\delta X_p. \tag{2-78}$$

Before proceeding with the above argument, one observation should be highlighted. It relates to the fact that Coats' formulation selects the set of primary variables of a gridblock according to its predominance of liquid or gas. In this sense, for this study, a gridblock can have a maximum of three types of states: biphasic, monophasic liquid (oily) and monophasic vapor (gaseous). Therefore, as Figure 2.8 shows, there are different sets of primary variables for each type:

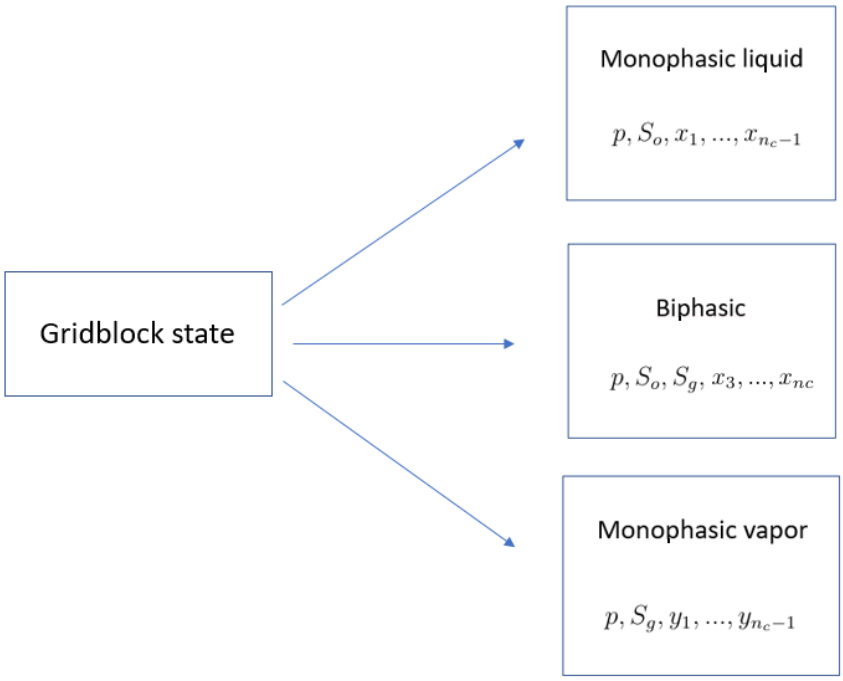

Figure 2.8: Different selection of primary variables depending on the state of the gridblock.

Since only one hydrocarbon phase (oil or gas) predominates in a single-phase gridblock (liquid or vapor), there is no reason to find variables related to the non-existent phase of this gridblock, nor to perform the thermodynamic equilibrium calculation in the form of fugacities. Taking the single-phase vapor cell as an example, it is only necessary to find the variables $p, S_g, y_1, ..., y_{n_c-1}$,

which means that it would not be necessary to perform the Gauss elimination procedure for this cell by proposing the following modifications to the diagonal submatrices, as shown in Figure 2.9:

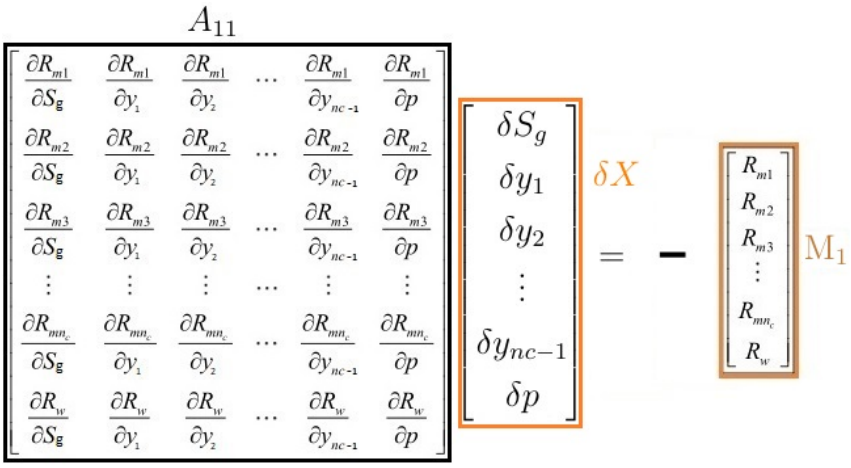

Figure 2.9: Diagonal submatrix of entire Jacobian matrix for monophasic vapor gridblock.

After presenting this observation, the focus is returned to the Gauss elimination procedure. Once having found δX_p and δX_s for the diagonal submatrices, it is necessary to find those for the off-diagonal submatrices. A similar method can be used to eliminate the secondary variables from the primary equations concerning the off-diagonal terms [3, 4], given by:

$$\begin{cases} A_{12} \longleftarrow A_{12} - B_{12} \cdot (D_2^{-1} \cdot C_2); \quad M_1 \longleftarrow M_1 - B_{12} \cdot (D_2^{-1} \cdot N_2) \\ \quad \vdots \\ A_{17} \longleftarrow A_{17} - B_{17} \cdot (D_7^{-1} \cdot C_7); \quad M_1 \longleftarrow M_1 - B_{17} \cdot (D_7^{-1} \cdot N_7) \\ A_{21} \longleftarrow A_{21} - B_{21} \cdot (D_1^{-1} \cdot C_1); \quad M_2 \longleftarrow M_2 - B_{21} \cdot (D_1^{-1} \cdot N_1) \\ \quad \vdots \\ A_{27} \longleftarrow A_{27} - B_{27} \cdot (D_7^{-1} \cdot C_7); \quad M_2 \longleftarrow M_2 - B_{27} \cdot (D_7^{-1} \cdot N_7) \end{cases} \tag{2-79}$$

It is important to emphasize that C and D matrices of a specific element do not take into account the primary equations, which contain the flux term, more specifically the transmissibility term. Therefore, there is no reason to have off-diagonal terms for C, D submatrices, nor to calculate them in (2-79) set. Another observation to be reemphasized is the neighborhood notation in the (2-79) set: the second subscript number denoting A, B submatrices refers to the neighborhood order. For example, A_{27} means the derivative of the second element residues due to the primary variables of its sixth existing neighbor.

Aiming to describe the method implementation, the flowchart for Coats' formulation is depicted in Figure 2.10 and subsequently presents a detailed description of each gridblock.

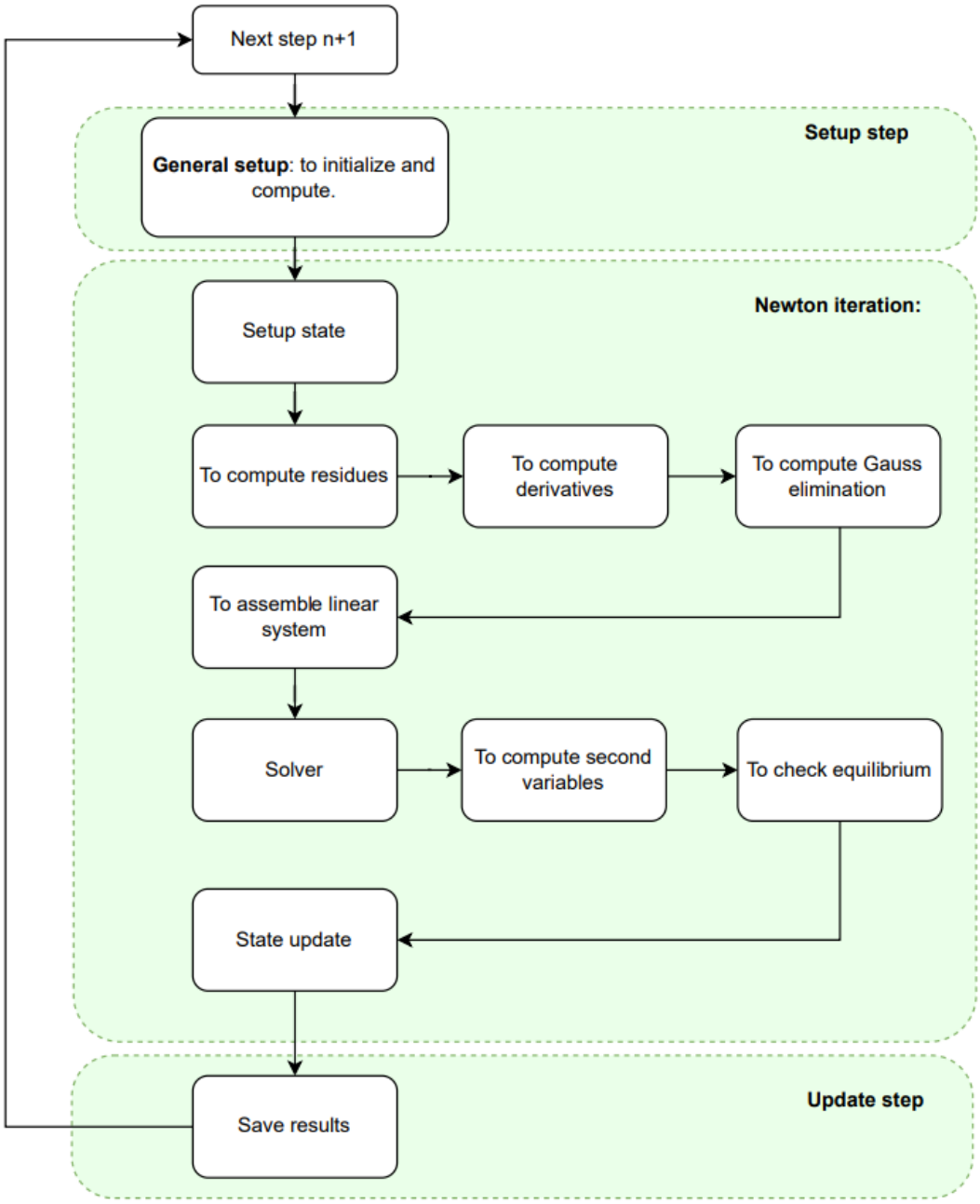

Figure 2.10: General implementation scheme of the Coats formulation for the simulator. Adapted from Fernandes [1].

Detailed description of Coats' flowchart.

SETUP STEP

–**General step:** Initialize and compute the requirements to go to the next time step;

NEWTON ITERATION

–**Setup state:** Compute the fluid, rock, and fluid-rock properties;

–**Compute residues:** Compute the primary equations (2-1), (2-2), and the secondary equations (2-33), (2-9), filling the M and N vectors for all cells;

–**Compute derivatives:** Compute the derivatives from the primary and secondary equations regard to primary and secondary variables. Assign the resulting values in the matrices A, B, C, and D for all cells;

–**Compute Gauss elimination:** Decouple the primary equations from secondary variables;

–**Assembly linear system:** Assemble the Jacobian matrix and the linear system equations;

–**Solver:** Solve the linear system for the primary variables;

–**Compute secondary variables:** To compute the secondary variables using Equation (2-78);

–**Check equilibrium:** Verify the mixture thermodynamic equilibrium for each cell and compute the Flash calculus for unstable cells;

–**State update:** Update the reservoir states;

UPDATE STEP:

–**Save results:** Save the results from the current time step and go to the next time step.

2.3.4 Initial Conditions

At the beginning of the simulation, it is necessary to assign initial values to the reservoir variables. These variables will be used to define the type of mesh (e.g., block size, number of blocks for each direction, location of producer and injector wells) and to calculate reservoir dynamics (e.g. T, p, S_w, z_i). Once this information has been loaded, the following procedure is used for all gridbloks:

1. Assign initial pressures at a reference depth;
2. Assign initial water saturations;
3. Assign initial overall compositions z_i;
4. Assign initial component mole fractions in the oil and gas phases, x_i and y_i. A Flash calculation is performed at the initial pressure, temperature and overall compositions z_i for each gridblock. The resulting x_i and y_i are assigned to each gridblock. At the same time, the oil and gas phase densities, and ξ_o, ξ_g, and the molar volume of the liquid phase ν_l are calculated for each gridblock;

5. Assign initial oil and gas saturations as:

$$S_o = \frac{1 - S_w}{1 + \left(\frac{1}{\nu_l} - 1\right) \frac{\rho_o}{\rho_g}}, \tag{2-80}$$

$$S_g = 1 - S_o - S_w. \tag{2-81}$$

If necessary, steps 4-5 are repeated to achieve better initial equilibrium.

2.3.5 Solution of the nonlinear equations

Reservoir flow equations are a set of nonlinear equations typically solved using the Newton-Raphson method. This method involves computing a Jacobian matrix [**J**] and a right-hand side **R** at each iteration of the Newton iteration process, i.e.:

$$[\mathbf{J}]\boldsymbol{\delta X} = -\mathbf{R} \tag{2-82}$$

where $\boldsymbol{\delta X}$ is the vector with the increments of the unknowns. The residual vector $\mathbf{R_n}$ for cell n is:

$$R_n = \begin{bmatrix} R_{i_n} \\ \vdots \\ R_{n_{cn}} \\ R_{w_n} \end{bmatrix}, \text{for } i = 1,...,n_c. \tag{2-83}$$

The configuration of the reservoir segment within the Jacobian matrix [**J**] is depicted in Figure 2.11. Essentially, it comprises three arrays: $Diag$, $OffD_{upper}$ and $OffD_{lower}$. $Diag$ represents the diagonal section, where each entry corresponds to a single gridblock, aligned according to the cell list. $OffD_{upper}$ captures the sparse structure of the upper triangular portion, while $OffD_{lower}$ delineates the sparse structure of the lower triangular portion. Each entry within these arrays pertains to a connection, positioned as per the connection list. For instance, in the case of a connection between gridblocks **U** and **L** (**U**<**L**), $OffD_{upper}$ records the derivative of gridblock **U**'s equations concerning gridblock **L**'s variables, situated at (**U**, **L**) within the Jacobian matrix. Similarly, $OffD_{lower}$ located at (**L**, **U**) contains the derivatives of gridblock **L**'s equations concerning gridblock **U**'s variables.

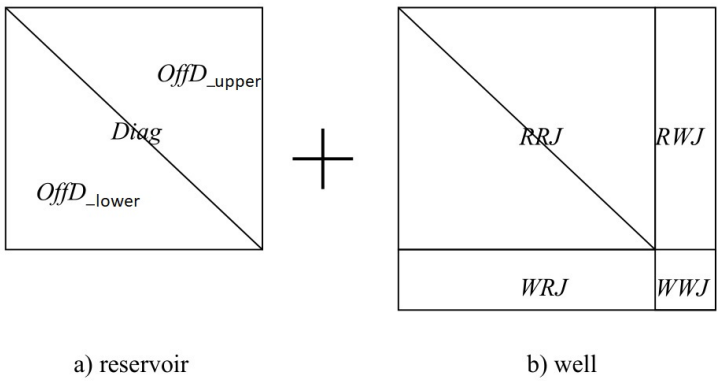

Figure 2.11: General structure of the Jacobian matrix. Taken from [3].

Regarding to the reservoir segment Jacobian calculations, the accumulation and flux components are distinctly handled. The accumulation phase iterates through the cell list, exclusively modifying terms within the Diag array. Conversely, the flux phase traverses the connection list, influencing all three arrays. In a multi-component system, each entry within these arrays represents a small dense matrix [3].

The general structure of the well segment in Jacobian for a single well is also illustrated in Figure 2.11b. The inclusion of wells requires extra rows and columns. Furthermore, alongside the reservoir variables, each well may introduce an additional variable, typically representing the well bottom hole pressure (BHP) p^W. As evident from Figure 2.11, the well segment of Jacobian comprises four arrays: **RRJ**, **RWJ**, **WRJ** and **WWJ**. **RRJ** means the derivatives of well terms in Equation (2-72) concerning the reservoir variables, exclusively located along the diagonal within the well blocks. **RWJ** portrays the derivatives of well terms in Equation (2-72) concerning the well variable. This additional well variable is indispensable for certain types of wells, such as those under constant rate control, requiring an additional well equation. **WRJ** and **WWJ** represent the derivatives of this additional well equation concerning the reservoir variables and the well variable, respectively. Typically, each entry in **RRJ** manifests as a small dense matrix. For **RWJ**, it is represented as a column vector, for **WRJ**, as a row vector, and for **WWJ** a single value. Each specified well will have its **RWJ**, **WRJ**, and **WWJ**.

For clarification, consider a reservoir represented by a mesh of NElem gridblocks. Also, these cells remain in monophasic vapor (gaseous) during all simulation time, which means that no oil phase appearance occurs in this case. In this particular case, there is a well defined by constant flow, which means that a new variable p^W has to be solved, so an additional column **RWJ**, a row **WRJ** and a single value **WWJ** appear in [**J**]. The order in which the cells are

numbered is such that the first cell to be numbered has six neighboring cells, and so on. In this sense the format of [**J**] is shown in Figure 2.12.

The superscript bracketed number in the denominator of the partial derivative indicates the numbering of the central cell while the last subscripted number in the denominator indicates the numbering of the neighboring cells (2,...,7), Q indicates Equation (2-56). To clarify the nomenclature used in Figure 2.12, select $\frac{\partial R_{m3}^{[1]}}{\partial y_{2,7}}$ as an example: it means the partial derivative of the residue equation linked to the third component i referring to the central cell 1, by the gas molar fraction of the second component i referring to the sixth neighbor of this central cell.

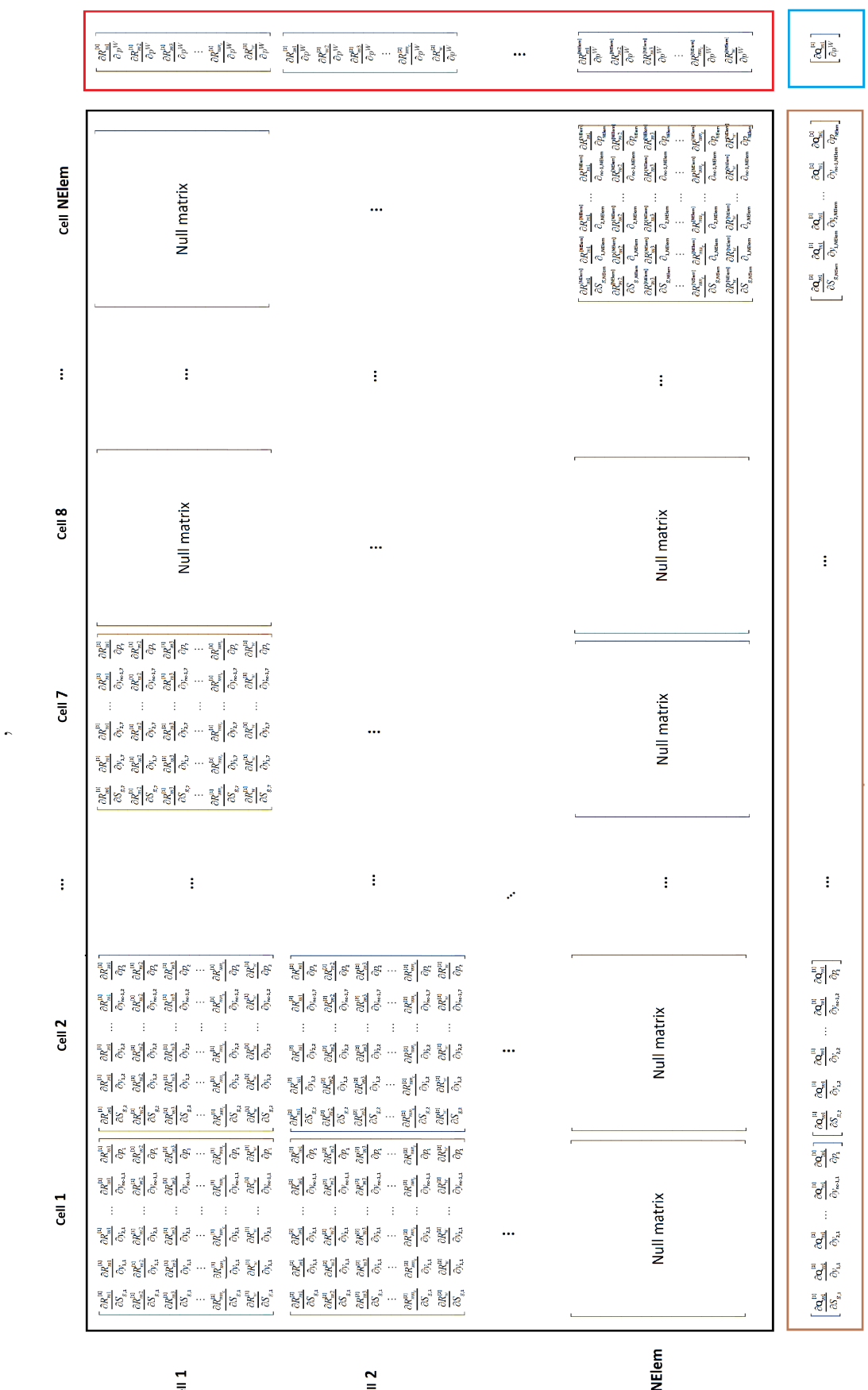

Figure 2.12: Full Jacobian [**J**] considering that all cells of the mesh are monophasic vapor (gaseous) state. The black, red, brown and blue boxes are RRJ, RWJ, WRJ and WWJ, respectively.

3
Basic aspects of the simulator

An important tradeoff of simulators is the relation between speed and accuracy. Simulators that use simple models reach convergence in less time but often sacrifice important details of the physical processes, resulting in inaccurate predictions [48, 70]. On the other hand, incorporating sophisticated mathematical models and algorithms that increase accuracy often requires more computational resources, consequently demanding more time to simulate.

Based on the aforementioned tradeoff, the GSIM reservoir simulator is based on the Topsim framework [6], which allows the interconnection of plugin tools. These plugins can execute specific functions of the simulation, enabling it to increase the accuracy of the model in specifics parts, also performing large-scale numerical analysis.

As emphasized in Chapter 1, this dissertation is a continuation of the work presented by Bastos [48], which details how GSIM works from a black oil perspective. The focus here is to propose modifications to GSIM to perform simulation based on the compositional approach. Thus, this chapter aims to explain how the theory and models discussed in Chapter 2 are implemented using plugin tools linked by the Topsim framework. In addition, this chapter will compare all the proposed modifications to the black oil model as they were explicitly discussed in Bastos dissertation [48].

3.1
Plugin-based framework

Developing a compositional reservoir simulator based on a monolithic architecture, which requires multiple codes or numerous executable files to manage each variation of the simulation process, would be impractical. A monolithic architecture would also present challenges in customization, integration, and comparison with other software [71]. The GSIM simulator was developed using the Topsim framework to avoid these issues.

The Topsim framework was proposed by Duarte [6] as a versatile tool for running complex numerical simulations. Its operation essentially depends on the integrated execution of plugins through APIs, short for application programming interface. Each of them defines all the functionalities required

for a specific application. As an example, Figure 3.1 shows the RockProps plugin.

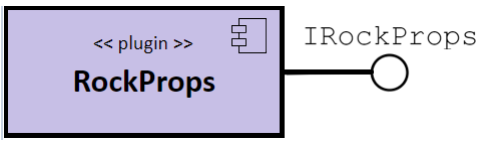

Figure 3.1: Schematic illustration of a plugin for the quantification of rock properties.

RockProps is responsible for calculating rock properties, like ϕ and $\bar{\bar{k}}$, using Equations (2-23) and (2-24), respectively.

Regarding the functionality of plugin, if it requires a specific task that is provided by another plugin, it is necessary to connect them using the required plugin interface. This plugin interface is a software element that lists all the functions that a particular plugin can provide. This functionality can be seen in the ReservoirCoatsFDM plugin, which is responsible to calculate partial derivatives of Equations (2-1)-(2-2). It is convenient to connect it to the RockProps plugin through its interface called IRockProperties, where the ReservoirCoatsFDM requires the ϕ new value after p perturbation, as shown in Figure 3.2:

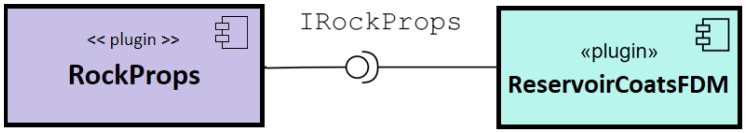

Figure 3.2: Schematic illustration of a connection between RockProps and ResevoirCoatsFDM plugins by IRockProps interface.

The Topsim framework functionality relies on two modules: the TopS data structure [72] and the Plugin Manager. The former is a compact topological data structure originally designed for representing finite element meshes. It manages efficiently storage space allowing for quick access all topological adjacency relationships. TopS is responsible for storing all crucial model data (attributes) shared among different plugins during analysis. In this way, plugins only handle specialized algorithms and their private data, while TopS acts as a bridge to update or retrieve any model data modified by any plugin.

In turn, the Plugin Manager is responsible to handle the dynamic loading, linking, and registration of plugins while the system is active. It relies on a Lua configuration script to facilitate communication between users and the application. This script serves as a tool for users to outline the setup

configuration of a simulation, detailing which plugins to load and how they should interact with each other. Figure 3.3 illustrates a simplified example of how Topsim works.

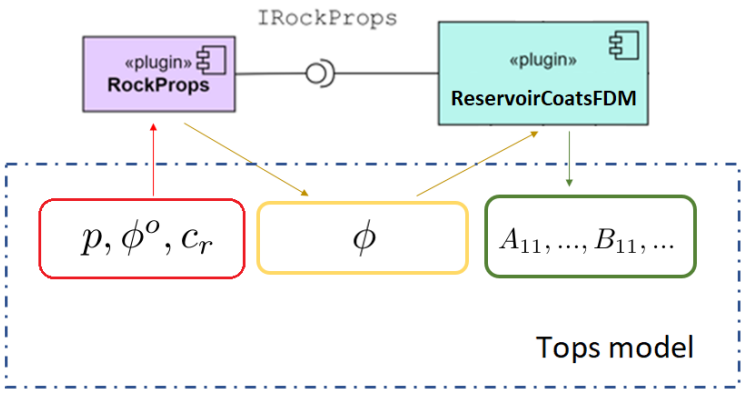

Figure 3.3: Schematic illustration of a Topsim framework application.

As previously mentioned, the reservoir simulator must be able to strike a balance between speed and accuracy, which makes maintenance (in the form of incremental modifications) vital [73]. Therefore, the entire Topsim structure has been designed as a software maintenance solution, offering the possibility to compare different formulations by modifying specific plugins. To emphasize the decision of choosing and maintaining Topsim for development of GSIM's compositional version, Figure 3.4 was constructed:

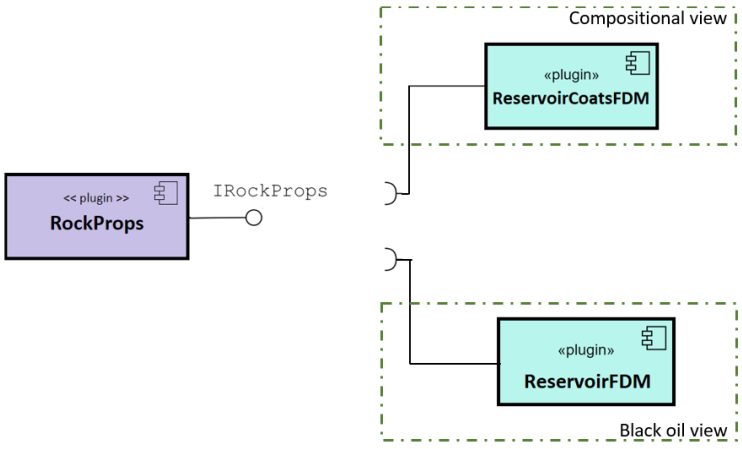

Figure 3.4: Topsim provides easy implementation in reservoir development.

3.2 GSIM's compositional version architecture

The compositional version of GSIM is divided as in the black oil version, into five main subgroups:

– Nonlinear solving (Figure 3.6);

– Input/Output (Figure 3.7);

– Geometry (Figure 3.8);

– Numerical solution (Figure 3.9);

– Linear solving (Figure 3.13),

which are gathered in Figure 3.5 and described in the following subsections.

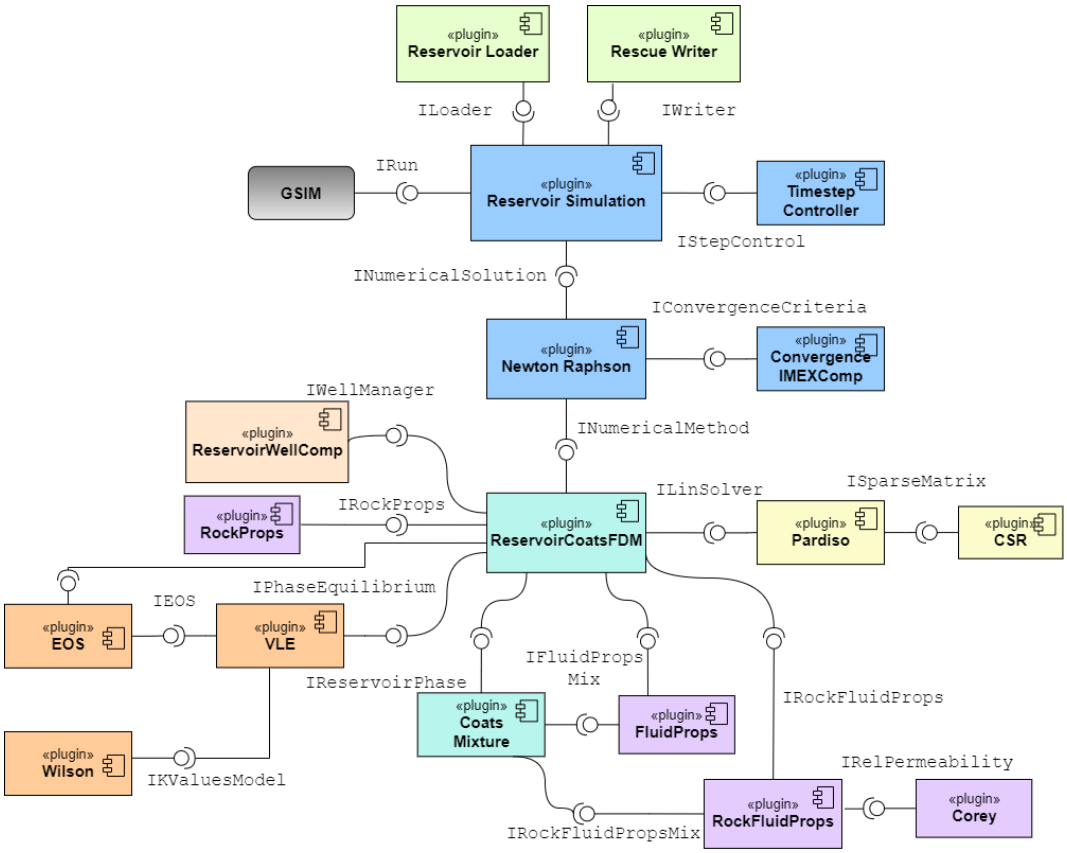

Figure 3.5: Selected plugins for GSIM's compositional version.

3.2.1 Nonlinear solving

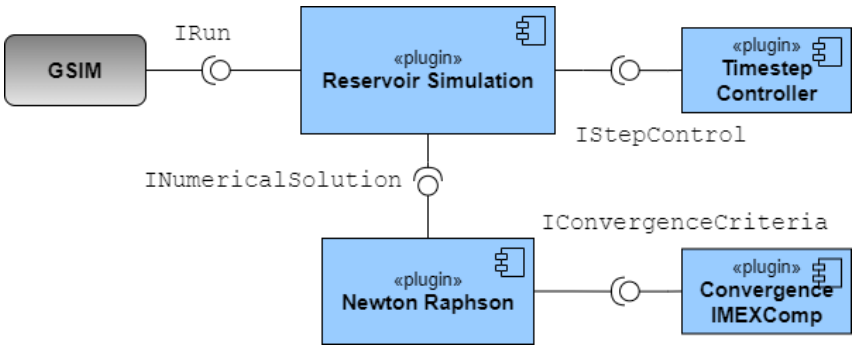

Figure 3.6: Selected plugins for nonlinear solving subgroup.

The numerical simulation starts with the nonlinear solving subgroup, which is responsible for executing an iterative nonlinear process, and after its convergence, advancing the time step. As performed in the black oil version, inside each time step several nonlinear iterations are done, where Equations (2-1) and (2-2) are linearized and assembled into [**J**]. After assembled it, they are solved using a direct linear solver (e.g., Pardiso). In compositional model, it is important to mention that there is an increased complexity when compared to the black oil model. This complexity becomes by the requirement to perform another loop of nonlinear iterations for the Flash routine within each nonlinear iteration previously discussed.

3.2.1.1 IRun and IStepControl

First, ReservoirSimulation plugin, defined by IRun interface, asks to fill TopS with initial conditions, in the form of attributes. After this process and once the mesh is generated, it starts counting the simulation time loop.

In the beginning of the loop, several actions need to be done in sequence: TimestepController plugin, defined by IStepControl, is responsible to set the size of the time step, while the INumericalMethod interface configures the equations relevant to the current time increment. The details of these setup procedures within each plugin will be discussed in later subsections of this chapter. Following this setup phase, the plugin will require INumericalSolution to find the solution for current time step.

Once convergence of the nonlinear iteration has been achieved, INumericalMethod proceeds to update X_p and X_s accordingly. The requirement order of these checks is facilitated by the cooperation of other plugins. At each time step iteration, INumericalMethod resets the variables values of the current step

to those of the previous step, effectively providing a initial guess for INumericalMethod. Before moving on to the next time step, ReservoirSimulation tasks RESCUEWriter plugin, which is responsible to write the results to designated output file.

3.2.1.2 INumericalSolution and IConvergenceCriteria

The objective of NewtonRaphson plugin, defined by INumericalSolution, is to determine the solution to the nonlinear partial differential equations within a specified time step. Its iterative approach starts with the cooperation of the ReservoirCoatsFDM plugin, which requests the CoatsMixture, FluidPropsMix, RockProps, and ReservoirWellComp plugins to compute mobilities, relative permeabilities, porosities, among others attributes. These properties are necessary to obtain the matrices and vectors **A**, **B**, **C**, **D**, **M** and **N**. Once these vectors and matrices have been obtained, ReservoirCoatsFDM performs the Gauss elimination procedure for all elements (only necessary if the element is biphasic) and stores its results in [**J**].

The next step after got [**J**] assembled, is solving Equation (2-82) in order to find the vector with the increments of the unknowns $\boldsymbol{\delta X}$. Therefore, the NewtonRaphson plugin proceeds to solve this global matrix linear system for each iteration by indirectly calling Pardiso, which is a parallel direct solver for large symmetric and non-symmetric matrices developed by Intel Corporation [74].

To determine whether the present iteration solution obtained from Pardiso converges or not, the CovergenceIMEXCOMP plugin is called to verify if the residual meets the pre-defined tolerance criteria[1], which are

- The residuals of the scaled mass balance equations are below 1×10^{-1}. The equations are normalized by the accumulation terms, which are the component masses divided by the time step size;
- The residuals of the scaled phase equilibrium relations are below 2×10^{-2}. These residuals are normalized by the component fugacities in the gas phase;
- The maximum normalized pressure change (absolute pressure change divided by the average reservoir pressure) is less than 1×10^{-4};
- The maximum change in saturation is less than 5×10^{-3};
- The maximum absolute change in component mole fraction is less than 1×10^{-3}.

[1]The convergence criteria values are taken from Cao et al [66].

In an event of non-convergence iteration, ReservoirCoatsFDM is called to reconstruct [**J**] with the incremented results of $\boldsymbol{\delta X}$ until reaches an iteration convergence.

Finally, having obtained the solution step from the system, ReservoirCoatsFDM is instructed to update X_p and X_s.

3.2.1.3 Input/Output

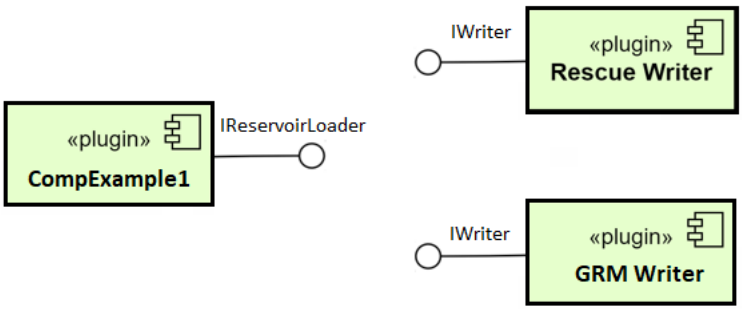

Figure 3.7: Selected plugins for input/output subgroup.

In order for GSIM to run simulations and display their results, it is necessary to import the input data and export the output data properly. This subgroup, composed by loader and writer tools, is responsible to deal with this process.

IReservoirLoader interface has the function to assimilate the input data given by the user (as described in Section 2.3.4), which are consequently loaded in the form of attributes. Also, in TopS the model is already converted to the adopted unit system. The plugin responsible to execute these functions in the compositional model is denominated CompExample1. The main difference between them and the ReservoirLoader presented in black oil version, is the treatment of the specific data from compositional model as z_i, μ_i, T_{c_i}, V_i, N_i, and ξ_α^i.

Additionally, the writer tool is designed to write all the data derived from the numerical simulation into an exportable output file. This exportable file is adapted to be visualized by GERESIM (a graphical interactive system for reservoir simulation management and geomechanics developed by TECGRAF in collaboration with Petrobras). In this sense, the GRMWriter and RESCUEWriter plugins were the same when is compared to the black oil and compositional version.

3.2.2 Geometry

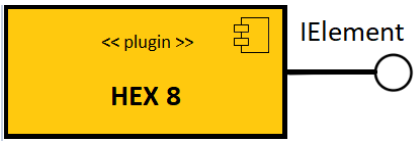

Figure 3.8: Selected plugin for geometry subgroup.

During the simulations, the properties related to the geometry of the element are occasionally required (e.g., the case of V_b when quantifying Equations (2-1) and (2-2)). The plugin responsible for providing this type of information is the same of the black oil version, HEX8, which is based on the classical 8-node hexahedron.

3.2.3 Numerical solution

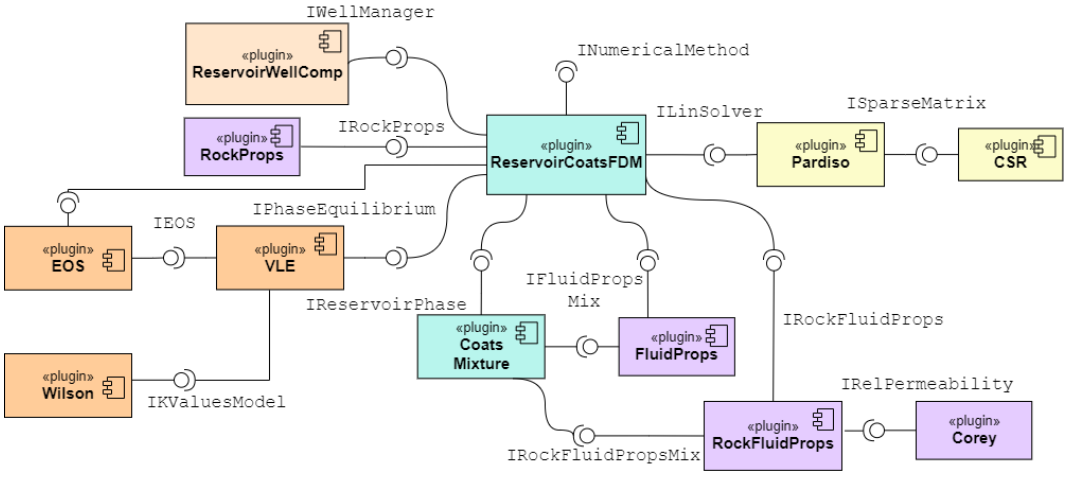

Figure 3.9: Selected plugins for numerical solution subgroup.

The numerical solution subgroup is considered the core of the compositional implementation, which deals with all discussion presented in Chapter 2. Consequently, this plugin arrangement is different from the black oil version.

3.2.3.1 INumericalMethod

The INumericalMethod interface defines the ReservoirCoatsFDM plugin whose main purpose is to assemble the global matrix linear system to be solved

by the linear solving subgroup. To achieve its objective, ReservoirCoatsFDM is used to specific which plugins are responsible for the calculation of reservoir properties, primary equations, secondary equations, and the Flash behavior.

The operation of ReservoirCoatsFDM can be divided into four main stages: simulation initializing, setup executing, global matrix linear system solution and variables updating .

The first stage takes place once at the start of the simulation and consists of obtaining the attributes required to start the numerical simulation. In this case, a first call is made to the RockProperties plugin to calculate the height difference between each element and its neighbors ΔH, the geometric factor of the transmissibilities T^g, the porosity ϕ, and the bulk volume V_b of each element. The second call is to the ReservoirWellComp plugin requiring the parameters needed to quantify the well terms q_i and q_w, which are, the constant well index WI, the productivity/injectivity index J_ω and the mobility factor λ_0. The third call is to the VLE plugin to run the initial Flash routine to find x_i and y_i associating them for each gridblock. All the calculation details performed by these plugins are described in their respective subsections. Finally, we need to allocate memory for the local matrices and vectors **A**, **B**, **C**, **D**, **M**, and **N** of each element, as well as for the global Jacobian matrix [**J**] (i.e., Figure 2.12). Since it is not known a priori which specific cases the user wishes to run in GSIM, the memory allocation procedure must be done dynamically (i.e., according to the input data provided, considering that all cells are set to the FI solution method for the compositional version). To support the creation of local matrices and vectors, the C++ template library Eigen has been added to the simulator. This addition was made because Eigen is versatile enabling to work with different sizes of matrices, from small fixed-size matrices to sparse ones. In this sense, Eigen is practical for providing different templates for mathematical operations such as calculating the inverse matrix, polynomial solvers, and matrix multiplications. Also, using Eigen, it is simple to create a vector of matrices or the [**J**], shown in Figures 2.7 and 2.11, respectively. It is only necessary to set their number of rows and columns according to the number of components i, flow rate specified wells, and number of cells given by the initial condition of the problem.

Once the local matrices, vectors and [**J**] sizes have been set up, the second stage done in ReservoirCoatsFDM begins. As mentioned in Section 3.2.1, for each nonlinear iteration, the NewtonRaphson plugin requests ReservoirCoatsFDM to execute its iterations and setup step function. Within this function, the VLE plugin is requested to perform the stability test procedure, which will indicate which cells have become unstable by the previous nonlinear it-

eration, and thus applying the Flash calculation to them. The CoatsMixture plugin is requested to compute the primary and secondary equations shown in the Equation system (2-53) using the values of the fluid and rock/fluid properties, and also q_i and q_w previously calculated by ReservoirWellComp. After finalizing this task, ReservoirCoatsFDM is able to quantify for each element the numerical derivatives of the primary and secondary equations in terms of X_p and X_s, and to assign their respective values to the previously allocated memory **A**, **B**, **C**, **D**, **M**, and **N**. Figure 3.10 illustrates how CoatsMixture, ReservoirWellComp and ReservoirCoatsFDM work together to get those tasks concluded.

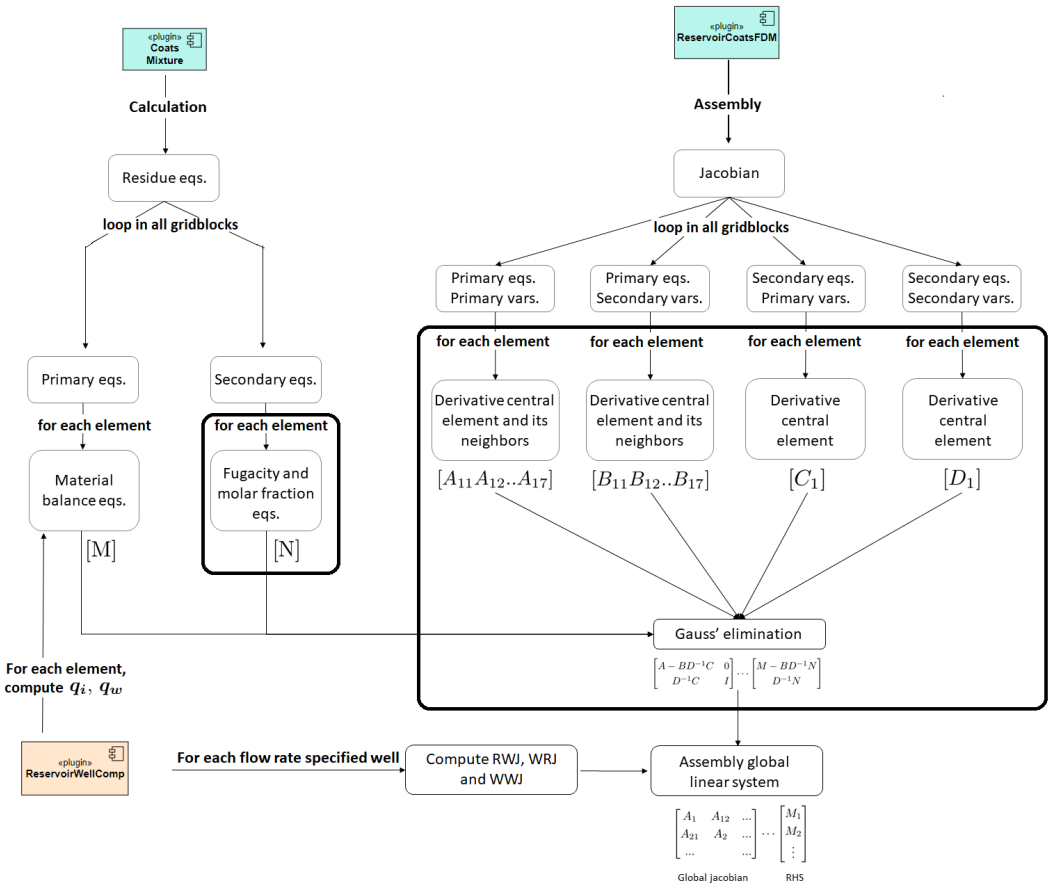

Figure 3.10: Synergy between ReservoirCoatsFDM, ReservoirWellComp and CoatsMixture plugins to assemble the global matrix linear system.

The operations circled and highlighted in black in Figure 3.10 endorse special attention to checking the state of elements. The cell state characterization procedure occurs after the Flash calculation and is detailed in Section 3.2.3.4. If the element is monophasic (i.e., liquid or vapor), it is not necessary to quantify **B**, **C**, **D** and **N**, nor perform a Gauss Elimination procedure and

it will be necessary to adapt the submatrices that compound matrix **A** (see Figure 2.7) according to its primary variables. Algorithm 1 shows a similar idea of implementation done in GSIM to deal with the above observation.

After performing the Gaussian elimination procedure described in Equations (2-74)-(2-77) for biphasic cells, the third stage begins. ReservoirCoatsFDM inserts the results in [**J**], requests to the ReservoirWellComp to calculate **RWJ**, **WRJ** and **WWJ**, assembles the global matrix linear system (Equation (2-82)), and requests the Linear Solver subsystem to find its solution.

Algorithm 1: Implementation of different routines depending on the element state

```
1  Loop elements up to the total number of elements Telem ;
2  for el = 0 to Telem do
       // All three states calculate the residue vector M
3      ComputeM(el);
4      switch el.state do
5          case Liq do
               // In case of monophasic liquid element
6              AssemblyA_liq(el);
7          end
8          case Vap do
               // In case of monophasic vapor element
9              AssemblyA_vap(el);
10         end
11         otherwise do
               // In case of biphasic element
12             ComputeN(el);
13             AssemblyA_biphasic(el);
14             AssemblyB(el);
15             AssemblyC(el);
16             AssemblyD(el);
17             GaussElimination(el);
18         end
19     end
20 end
```

Finally, the fourth stage begins, where for each nonlinear iteration, NewtonRaphson asks the ReservoirCoatsFDM to request CoatsMixture and ReservoirWellComp to update X_p, X_s and p^W (if present). After this update process, since x_i and y_i have new values, it is necessary to check if there is any cell that is not in thermodynamic equilibrium. Therefore, VLE is requested to perform a stability test on all biphasic cells, and consequently, a Flash routine on those that were marked as unstable. The nonlinear interactions end

when the convergence criteria are reached. At this point, ReservoirSimulation requests ReservoirCoatsFDM to update the variables related to the current time step, which is done by CoatsMixture and ReservoirWellComp copying the variables value from the step $n + 1$ to n.

3.2.3.2 IReservoirPhase

This interface was designed to calculate the primary and secondary equations (System of equations (2-53)). In this way, this dissertation developed a plugin called CoatsMixture to deal with a hydrocarbon mixture, which is opposed to what occurs in the black oil version (i.e., the latter uses one plugin for each single phase, which are, WaterComponent, OilComponent, and GasComponent).

As mentioned in the previous section, at the second stage of the ReservoirCoatsFDM, the CoatsMixture is requested to compute the attributes calculated by RockProperties, FluidPropsMix, RockFluidPropsMix, and ReservoirWellComp plugins. These requested attributes are used to fill **M** and/or **N** residue vectors of each element, already pre-allocated using Eigen's dynamic vector structure.

Once all the attributes have been calculated, the quantification of the primary and secondary equations is started. The former one is divided into three terms, accumulation, flux, and well, which are calculated separately for each component i and water components. It is important to emphasize that each calculation is done separately for each cell and inserted directly into its respective position in the element **M** residue vector. Special observation is needed for the flux term quantification: during the calculation of the λ_0, in order to check which cell is upstream in the looped pair (between the central element with each of its neighbor cells), CoatsMixture requests TopS for the value of Φ according to the criterion shown in Equation (2-55). Once got Φ and identified which cell is the upstream, CoatsMixture gets the k_{r_α}, μ_α and x_α^i values from this upstream element to compute λ_0. For the well term quantification, ReservoirWellComp is requested where its procedure details were previously discussed in its corresponding subsection. In turn, the secondary equations are computed and its values inserted in the element **N** residue vector. In this implementation, the position of each equation in **M** and **N** residue vectors follows the order presented in Figure 2.6.

3.2.3.3 IWellManager

Since each well and its completions have their own specific characteristics that define its physical operations and dynamic behavior, they are represented as data structures which are updated and stored in TopS. Figure 3.11 shows the similar Well and Completion data structures used in the GSIM simulator. As it is done in the black oil version, three types of information are needed to define each well structure: geometric/physical properties, status, and flow rate specification. The first group includes the well radius R_w, the geometric factor G_f, the skin factor s, and the circle factor c. In the second group are defined the name and the ID of each well (for identification), whether the well is open or not (its status), its type (injector or producer), and the number of completions. With the number of completions information, a vector called List of Completions is created, where all existent completion IDs belonging to this specific well are dated. In the third group, the well operational condition is defined. For injection wells, either the reference p^W (i.e., the well bottom hole pressure), the water or the gas flow rates can be used to specify its operational condition. To specify producer wells, the p^W, the water, gas, oil and the total flow rates can be used.

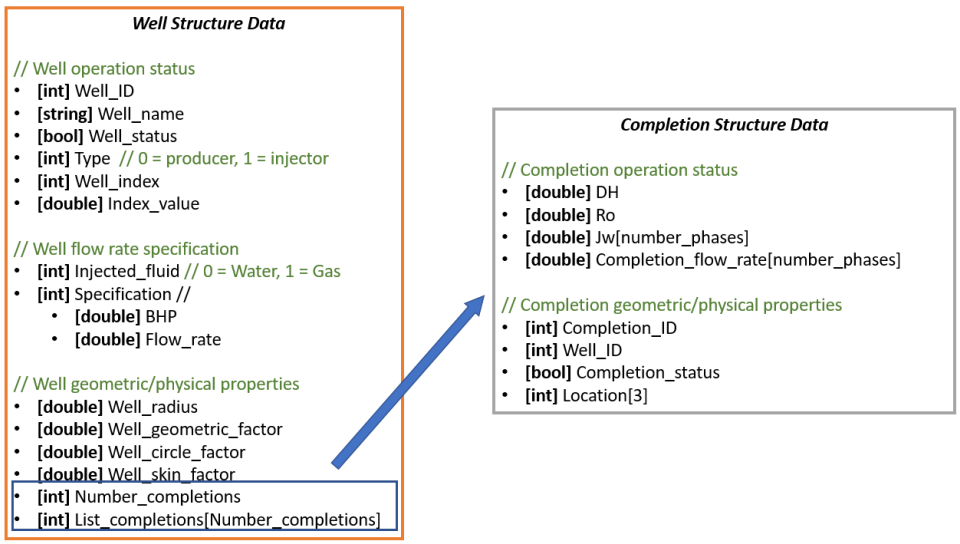

Figure 3.11: Similar Well and Completion data structures used to fill wells and its completions information.

Regarding the completion data structure, two types of information are provided: the operation status and the geometric/physical properties. The first one includes the height difference (DH) between the current completion and the reference depth of the analyzed well, the equivalent well radius R_o, the productivity/injectivity index J_ω for each fluid present in the completion, and

the flow rate of each phase (q_w, q_o and q_g). Finally, the status data group contains information on whether the completion is closed or open, the ID and location (in x, y and z directions) of the completion analyzed, and the well ID to which it belongs.

The IWellManager interface is responsible for calculating all attributes concerning to well conditions using the structures above presented, which defines ReservoirWellComp plugin. The first call of ReservoirWelComp is done by the first stage of ReservoirCoatsFDM for the well data initialization. This initialization loops through the wells and their completions, calculating r_{eq}, WI and the height difference between each cell containing the completion and the reference depth of the respective one.

The second call to ReservoirWellComp is done by the second stage of ReservoirCoatsFDM for calculating, at each nonlinear iteration, the J_ω for each completion and then to calculate the flow rate q_i for each component using Equation (2-56). In this regard, ReservoirWellComp obtains from TopS the values of the molar density ρ_α previously calculated by FluidPropsMix and checks which phases α are present in the analyzed wellbore in order to sum them to account for q_i. Once this task has been completed, ReservoirCoatsFDM is allowed to quantify **A**, **B**, **M** that the well term is presented. In addition, when [**J**] is being assembled, ReservoirWellComp has already quantified the number of flow rate specified wells in order to add additional values for **RWJ**, **WRJ** and **WWJ**.

Finally, the third call is concerned to update the well variables between nonlinear iterations and after its convergence.

3.2.3.4 IPhaseEquilibrium,IEOS

This subset of interface is exclusive to the compositional version as it works with thermodynamic equilibrium. In addition, this equilibrium makes the simulator much harder to implement, especially to the Flash behavior, which costs almost up to 30% of the total running simulation time [3]. It consists of the VLE and EOS plugins, which are defined by the IPhaseEquilibrium and IEOS interfaces, respectively. VLE, short for Vapor-Liquid Equilibrium, is responsible for carrying out the stability test and Flash routine discussed in Sections 2.2.3 and 2.2.4. To carry out these two procedures, gathered in UpdateEquilibrium function, it must call EOS (short for Equation of state) to quantify f_α^i using either Soave-Redlich-Kwong or Peng-Robinson (1976) approaches, and Wilson method to calculate K_i defined by the IKValuesModel interface using Equation (2-37).

At the simulation start, all cells are intentionally marked as unstable biphasic state in the stability test for two reasons: firstly, to determine x_i and y_i values of each cell from z_i given as initial condition using Flash procedure, and secondly, to avoid incorrect initial cell states given by users which could lead to further fatal error cascade during the simulation process. After the nonlinear iterations, the UpdateEquilibrium function from the VLE plugin is called to determine which cells have changed their state, and then, to apply the Flash calculation to these cells. Figure 3.12 clarifies the way VLE, Wilson, and EOS work together to deal with the stable thermodynamic equilibrium of all cells and shows the VLE's UpdateEquilibrium function procedure described above.

Finally, in the third stage of the ReservoirCoatsFDM, VLE's UpdateEquilibrium function is called to detect elements that occasionally switched state after X_p and X_s update.

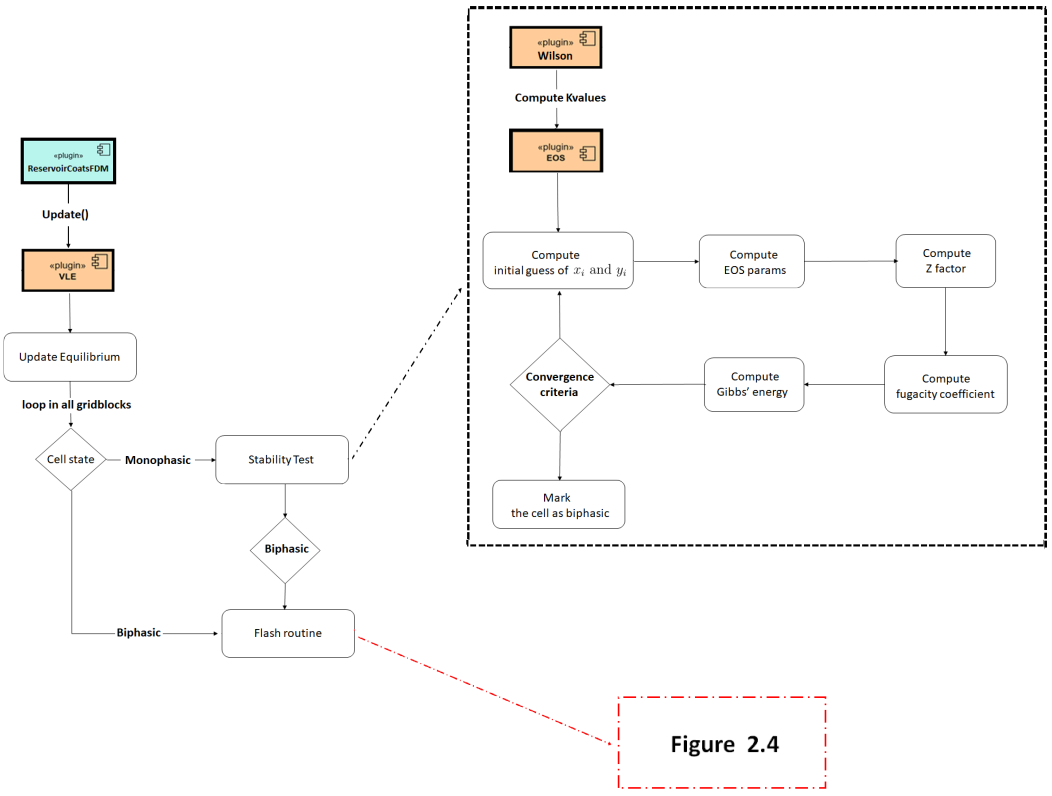

Figure 3.12: Synergy between VLE, Wilson and EOS plugins to verify stable thermodymamic equilibrium of each gridblock.

3.2.3.5 Auxiliary interfaces

This subsection details specialized interfaces by providing information of interfaces discussed previously. In this group, the plugins IRockProps, IFluidPropsMix, IRockFluidPropsMix, and IRelPermeability are discussed.

The properties of the rock are the same used in the black oil version. Therefore, the RockProps plugin defined by IRockProps interface in black oil version is maintained for compositional version. It is called once by ReservoirCoatsFDM in its first stage to provide initial values of ΔH, V_b, and T^g. Considering ΔH, it is requested HEX8 to give the centroid of each element, and further, is computed the height difference between each cell against its neighbors, subtracting its centroid position. Regarding the calculation of the bulk volume V_b of each element, it is also obtained requesting HEX8, which consults its length, width, and height. For T^g, Equation (2-55) is used with the length Δ, where the area perpendicular to the flux A given by HEX8 and the permeability for each direction k are given by TopS. Also, the element porosity ϕ is calculated in this plugin using Equation (2-23) which is called in the first two steps of ReservoirCoatsFDM (i.e., when calculating its initial value and its perturbed value for numerical derivative quantification).

On the other hand, the IFluidPropsMix interface is reserved to calculate fluid properties. Its defines the FluidPropsMix plugin, which is called to compute ξ_α, N_i, μ_o and μ_g using Equations (2-5), (2-10), (2-11) and (2-12), respectively. It is importat to notice that these computations are called twice for each variable (i.e., for their initial values and for their numerical derivatives). After each nonlinear iteration, CoatsMixture is also used to calculate the new values of z_i once S_α, x_i, and y_i have been updated.

Finally, IRockFluidProps and IRelPermeability interfaces are linked to calculate the initial and perturbed values of $k_{r\alpha}$. While the Corey plugin provides the values of $k^o_{r\alpha}$, S^o_α, and t_α, RockFluidProps uses them to calculate $k_{r\alpha}$ according to Equation (2-27).

3.2.4 Linear solving

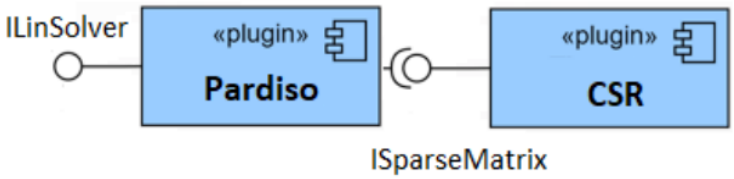

Figure 3.13: Selected plugins for linear solver subgroup.

One of the many challenges in developing a compositional simulator is to represent the dynamic behavior of the reservoir in a mathematical approach, which was discussed in detail in Subsection 3.2.3. After this step, another main challenge is solving the global matrix linear system derived from multiphase flow for each nonlinear iteration. Its difficulty consists of dealing with high sparse matrices whose format can vary for each case analyzed. Therefore, a smart way to express these sparse matrices without storing their zero value positions is crucial for memory allocation. Also, it is important to facilitate linear solvers by reducing the number of operations to be performed, which consequently affects the simulator performance.

3.2.4.1 ISparseMatrix

CSR plugin, short for Compressed Sparse Row, was developed for this purpose, it consists of using three arrays: Row Pointer, Column Index, and Values to store a sparse $(m \text{ x } n)$ $\boldsymbol{A}$ matrix. For the sake of illustration, Equation 3-1 shows an example:

$$\begin{aligned} [\mathbf{A}] &= \begin{bmatrix} 17 & 0 & 0 & 22 \\ 0 & 3 & 1 & 0 \\ 17 & 0 & 0 & 5 \end{bmatrix} \\ \text{Values} &= [17,\ 22,\ 3,\ 1,\ 17,\ 5] \\ \text{Row Pointer} &= [0,\ 2,\ 4,\ 6] \\ \text{Column Index} &= [0,\ 3,\ 1,\ 2,\ 0,\ 3] \end{aligned} \quad . \tag{3-1}$$

The Values vector contains, as expected, the matrix nonzero entries presented in [**A**] following the order, left to right, and from the top to bottom. In turn, the Row Pointer array contains the information where each line of $\boldsymbol{A}$ begins in the Values vector. According to the example, 17 and 3 are starter of

their respective row, therefore, Row Pointer will get their numbering position within Values array. Finally, Column Index has the same size as Values vector and holds the column index of each nonzero value.

3.2.4.2 ILinSolver

As mentioned in Section 1.1.6, there is two types of linear solvers: direct and iterative. The former is known to be highly competitive for small-size systems, while the latter is more efficient as the number of system unknowns arises [21]. Four linear solvers are available in GSIM: Pardiso, Biconjugate Gradient Stabilized (BICGSTAB), Generalized Minimal Residual (GMRES), and Orthogonal Minimization (ORTHOMIN). The first is a direct type, while the others are iterative.

Once [**J**] has been assembled, ReservoirCoatsFDM requests one of the linear solvers available in GSIM to solve the large sparse global linear system of equations.

To conclude this chapter, Figure 3.14 lists all the interfaces and plugins available for both the black oil and compositional versions of GSIM.

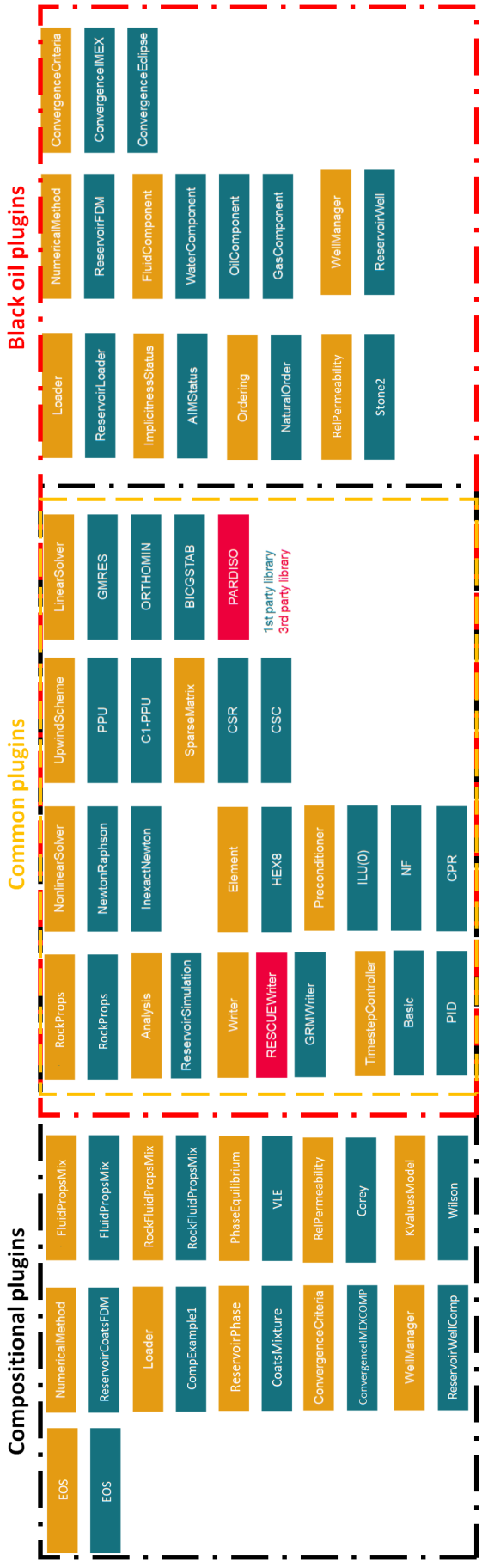

Figure 3.14: Summary of all interfaces and its plugins available for both black oil and compositional versions of GSIM.

4
Numerical Simulations

The analysis of the results are performed for test cases using the proposed GSIM's compositional version. The computer specification used for running the GSIM compositional version are detailed in Table 4.1.

Table 4.1: Machine configurations used.

Operating system	Windows 10 Pro Education (64-bit)
RAM	32.0 GB
CPU	Intel Core i7-3960 3.20GHz
Compiler	Intel C++ 21.4

4.1
Water Flooding

The first example consists of a heterogeneous, anisotropic oil reservoir characterized by two components: C_1 and NC_{16}, whose composition and properties are detailed in Tables 4.2 and 4.3. For this case, the binary coefficients κ_{ij} are considered zero. Although this example is a triphasic test case, there is no gas phase present in the elements at the initial simulation condition. Also, it is important to mention that this example has been studied by Santos [4].

Table 4.2: Overall molar fraction composition for Water Flooding test.

Component	$\mathbf{z_i}$
C_1	0.1
NC_{16}	0.9

Table 4.3: Component properties for Water Flooding test.

Properties	$\mathbf{C_1}$	$\mathbf{NC_{16}}$
T_c [K]	190.60	734.50
P_c [atm]	45.40	17.15
Critical volumes $[m^3/\text{kmol}]$	$9.90\text{x}10^{-2}$	$8.35\text{x}10^{-1}$
Acentric factor w_i	$8.00\text{x}10^{-3}$	$6.84\text{x}10^{-1}$
Molar mass [g/kmol]	$1.60\text{x}10^{1}$	$2.22\text{x}10^{2}$

Once the components properties are given, it is necessary to describe the wells information. A five-spot configuration is a common layout used in reservoir simulations, which consists of five wells: four producers in each corner and one injector in the middle [1] (see Figure 4.1). Due to its symmetry, it is common practice to analyze only one quarter of the domain (yellow region of the figure) to avoid unnecessary computational cost. Therefore, for this dissertation, only one quarter is simulated, where water is injected through the well injector.

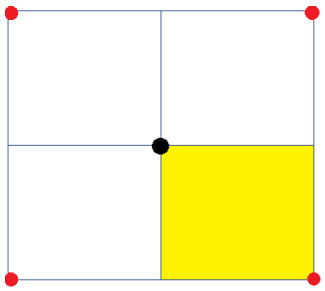

Figure 4.1: A five-spot reservoir configuration. The red and black circles are producers and injector wells, respectively.

Initially, the reservoir contains only oil and water phases. To calculate μ_α, it is used the first model (Equations (2-11) and (2-12)) presented in Section 2.1.2, which employs a sum of μ_i. Regarding the calculation of relative permeability, Corey's model was chosen and the parameters used are detailed in Table 4.4. In addition, Table 4.5 lists the remaining reservoir data used for this case.

Table 4.4: Corey's model parameters.

	Water	**Oil**	**Gas**
End point relative permeability	0.3	0.75	0.9
Residual saturation	0.25	0.2	0.0
Exponent of relative permeability	2.0	2.0	2.0

Table 4.5: Fluid composition for Water Flooding test.

Property	Value
Reservoir length, width and thickness [m]	609.60, 609.60 and 61.00
Porosity	0.30
Reservoir temperature [°C]	18.33
Rock compressibility [kPa^{-1}]	5.80x10^{-7}
Water compressibility [kPa^{-1}]	4.35x10^{-7}
Water viscosity [Pa.s]	8.00x10^{-4}
Water density [mol/m^3]	55.55x10^3
Initial water saturation	0.25
Initial reservoir pressure [MPa]	20.68
Water injection rate [m^3/s]	9.20x10^{-3}
Producer bottom hole pressure [MPa]	20.68

Finally, to analyze the influence of the hydrostatic force in the reservoir, the first example is divided into two subcases: a two-dimensional case and a three-dimensional case.

4.1.1 Bidimensional reservoirs

In order to verify the magnitude of the grid effect in the reservoir simulation, it is used three types of grid refinement: 20x20, 30x30 and 50x50. The position of the producer and injector wells within the selected grids is the same as in the yellow region shown in Figure 4.1. All three subcases have a time simulation of 7300 days (i.e., 20 years) and use a constant value of 2000 md for permeability.

Figures 4.2 and 4.3 show the pressure p and oil saturation S_o distribution along the cell mesh obtained by GSIM and CMG-GEM in the end of simulation of 20x20 subcase. Figures 4.4 and 4.5 show the pressure and oil saturation fields for the three simulated subcases, where it can be noted that the pressure and oil production rate values tend to stabilize, as expected.

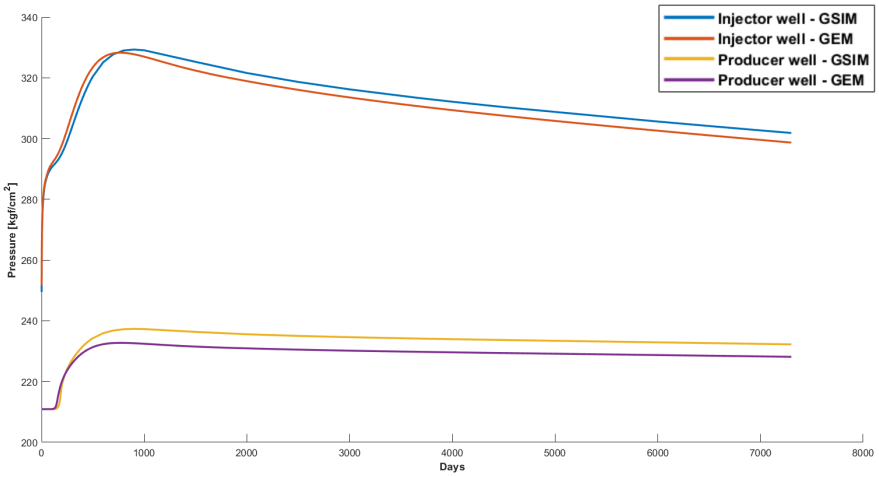

Figure 4.2: Pressure curve comparison between GSIM and CMG-GEM at injector and producer wells, using a 20 x 20 grid.

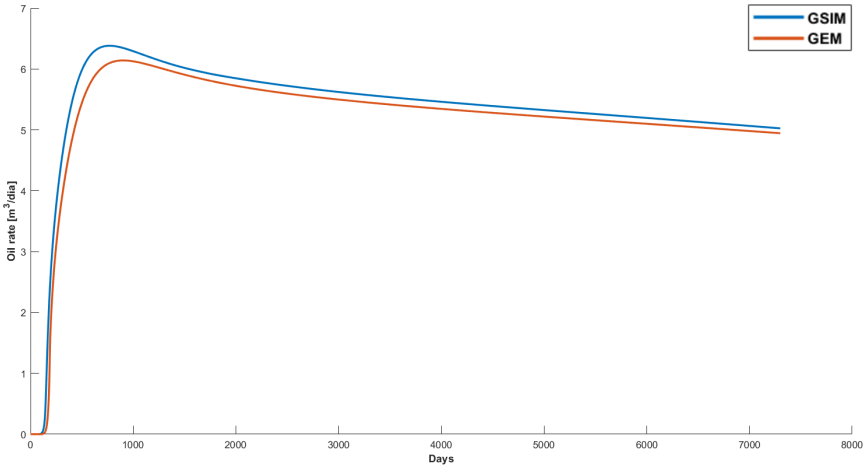

Figure 4.3: Oil production rates comparison between GSIM and CMG-GEM at injector well, using a 20 x 20 grid.

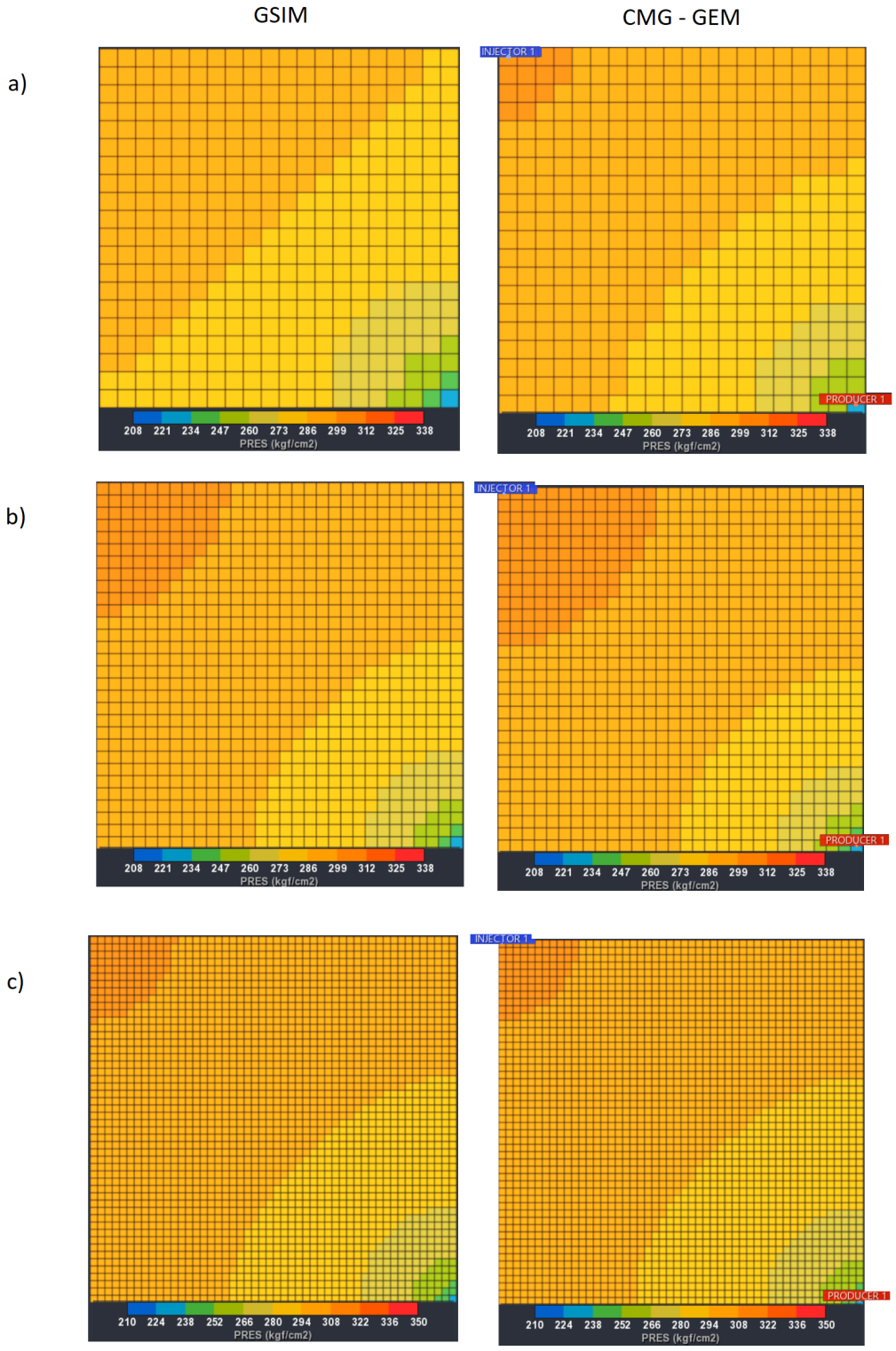

Figure 4.4: Pressure p field at 7300 days simulated with 20x20 (a), 30x30 (b) and 50x50 (c). The first column result is obtained from GSIM and the second from CMG-GEM.

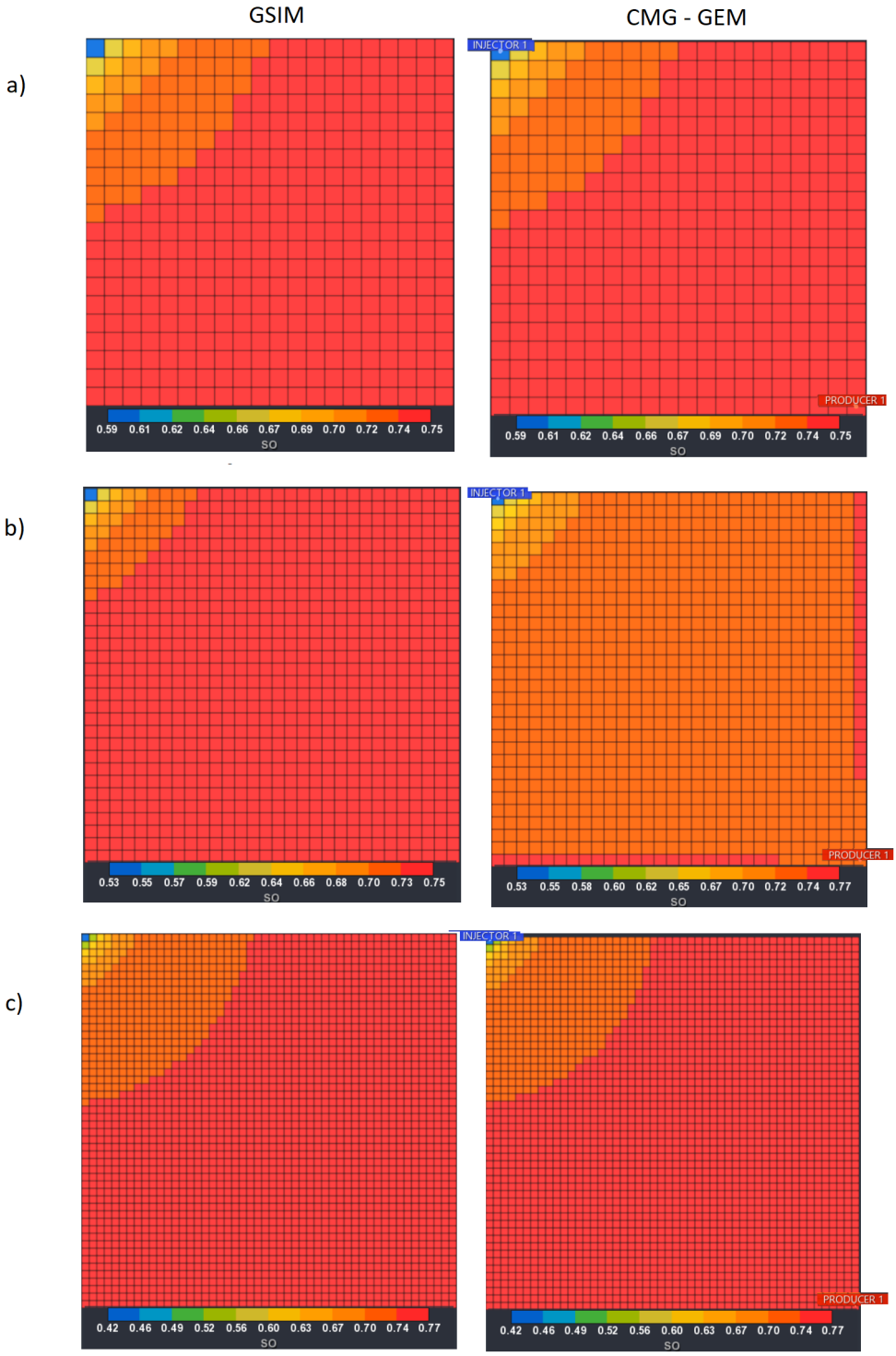

Figure 4.5: Oil saturation S_o field at 7300 days simulated with 20x20 (a), 30x30 (b) and 50x50 (c). The first column result is obtained from GSIM and the second from CMG-GEM.

As expected, the region around the injector well is the one with the widest ranges of pressure p values. As we move away from it, the pressure wave begins

to soften until reach the point where the producer well is located. The same reasoning can be applied to oil saturation S_o values. In this respect, it can be seen that the behavior of p and S_o presented by the two simulators turns out to be very similar. The last observation is intended for the cell states. At the start of the simulation, it was mentioned that all the cells in the reservoir contained only water and oil phases. The gas phase does not appear in any of the cells during the simulation. To illustrate this aspect, Figure 4.6 shows the state of the cell containing the producer well during the simulation. The cell state values vary between 0 (monophasic liquid), 1 (biphasic), and 2 (monophasic vapor).

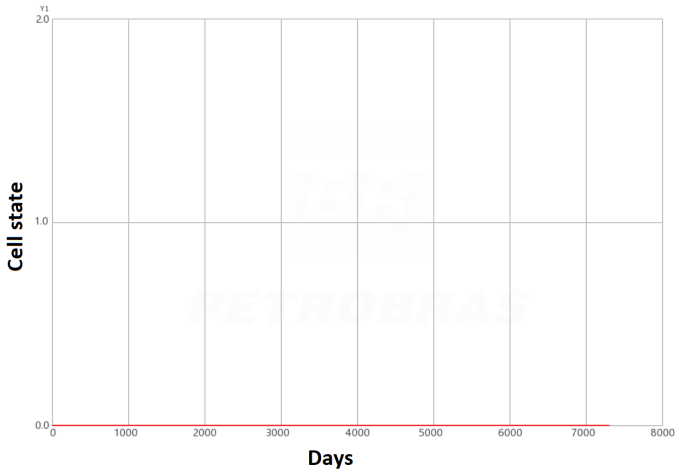

Figure 4.6: Cell state at producer well during the simulation for all the three subcases.

One observation from Figure 4.6 is that no gas phase is formed during the entire simulation, which means there is no migration of components between hydrocarbon phases. To emphasize this fact, the oil molar fraction values of the C_1 and NC_{16} components from the bidimensional 20 x 20 elements subcase were plotted and shown in Figure 4.7. The component C_1 had its oil molar fraction a constant 0.1 value next to NC_{16}, the value of 0.9. Since Water Flooding test case has no presence of gas phase, the input overall molar fraction z_i equals to oil molar fraction x_i.

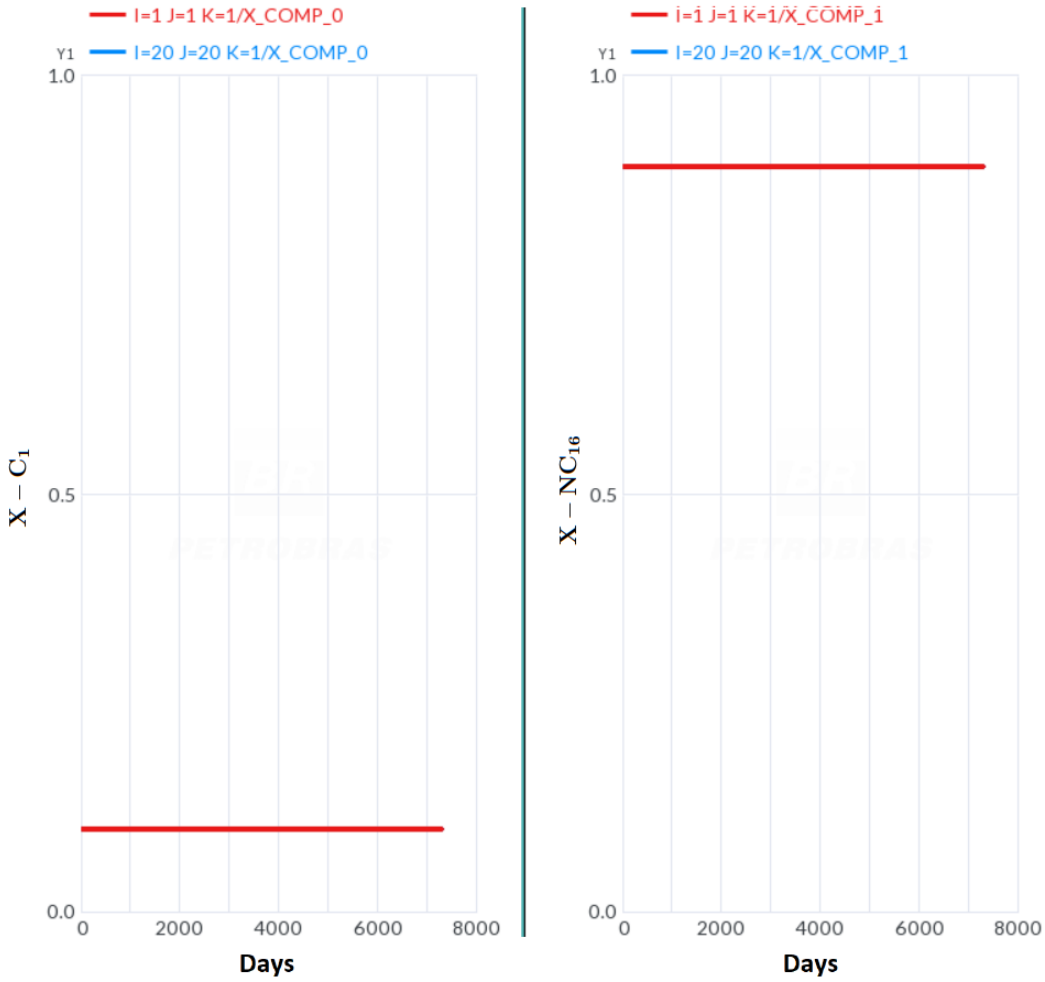

Figure 4.7: Oil molar fraction curves shown in the element where the injector (1, 1, 1) and the producer (20, 20, 1) well are located during the bidimensional subcase 20 x 20 elements simulation.

4.1.2 Tridimensional reservoir

This simulation aims to analyze if the results presented in the bidimensional subcases are similar to the tridimensional case. In this way, the permeability field in the x and y direction was given as the input parameter, shown in Figure 4.8. For the z direction, its values are equivalent to 10 % of the x direction values. The id positions of the injector and producer wells are $(1, 50, 3)$ and $(50, 1, 3)$, respectively. The simulation time of 7300 days (20 years) is maintained.

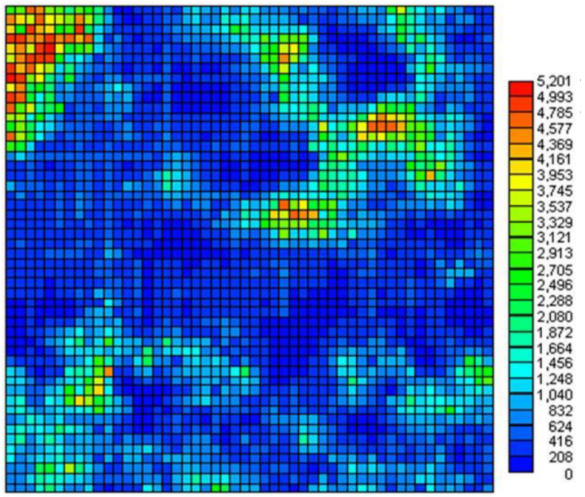

Figure 4.8: Horizontal absolute permeability in md. Taken from Santos [4].

The first result to be presented is the p pressure curve throughout the simulation, shown in Figure 4.9. It shows that, as in the previous subcases, the injector well has higher values than the producer. In addition, the values presented by GSIM and CMG-GEM are similar.

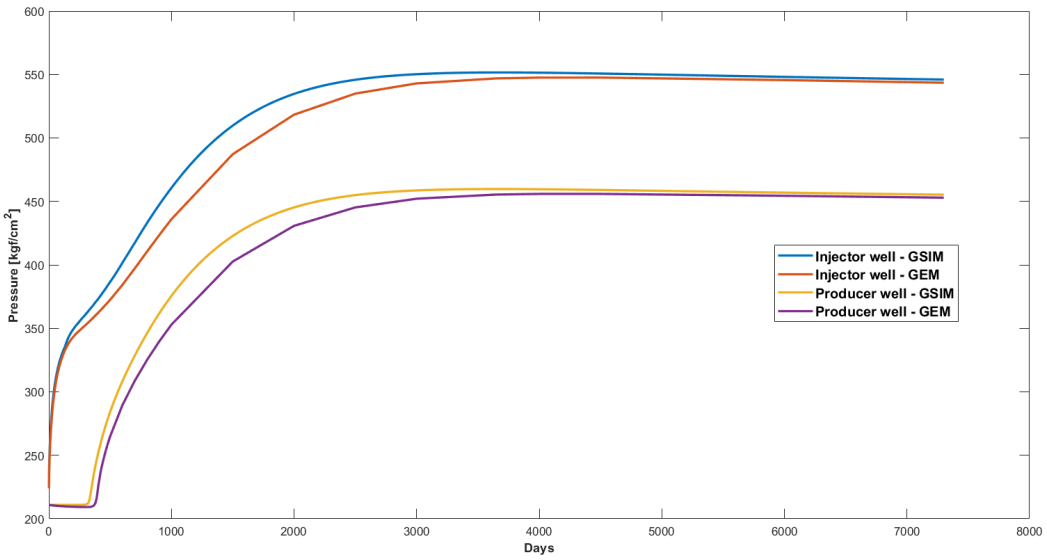

Figure 4.9: Pressure curve comparison between GSIM and CMG-GEM at injector and producer wells.

The next results shown in Figure 4.10 refer to the pressure fields and the three saturations.

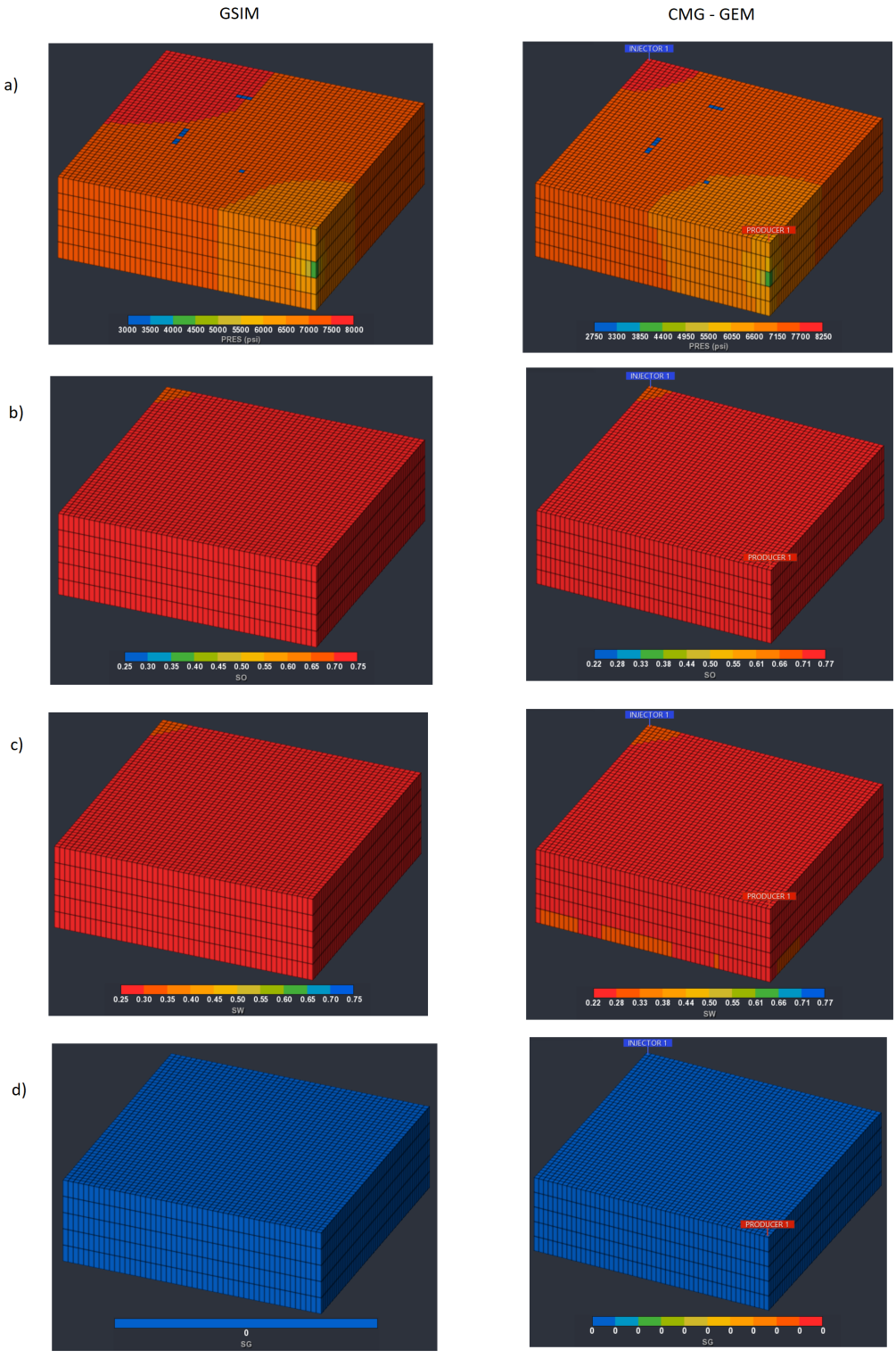

Figure 4.10: Pressure p (a), S_o (b), S_w (c) and S_g fields at 7300 days simulated. The first column result is obtained from GSIM, and the second from CMG-GEM.

A special observation is made in this three-dimensional case: for the

pressure field at the end of the simulation, as it approaches the producer well, the smoothing of the pressure wave in all three directions of the reservoir is apparent. The comments done in the previous subcases can also be extended for the behavior presented here.

4.2 Flash test

This example aims to emphasize the concept of the Flash procedure. In this sense, for the sake of simplicity, only five one-dimensional queued cells were selected for this 1000-day simulation in GSIM. All the input parameters adopted in Subsection 4.1.1 were maintained. The only exception would be that, for this case, there is no injector well. To force the flow from the first cell to the fifth (where the producing well is located), a 10% increase in the initial pressure value was input for all cells. Finally, for this example, the component NC_{16} was split into two components (NC_{16a} and NC_{16b}), both maintaining the properties values of component NC_{16}. Therefore, the input overall molar fraction is shown in Table 4.6.

Table 4.6: Overall molar fraction composition for Flash test.

Component	$\mathbf{z_i}$
C_1	0.01
NC_{16a}	0.19
NC_{16b}	0.80

The first result shown in Figure 4.11 refers to the pressure values in the five elements at the end of the simulation.

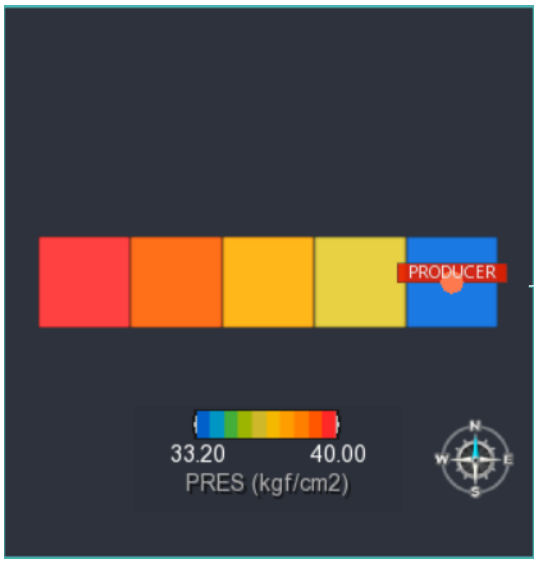

Figure 4.11: Pressure p field at 1000 days simulated. Obtained from GSIM.

In this example, it can be observed that the pressure decreases as we move from the first to the fifth element. Figures 4.12 and 4.13 show the behavior of the pressure and the three saturations during the simulation.

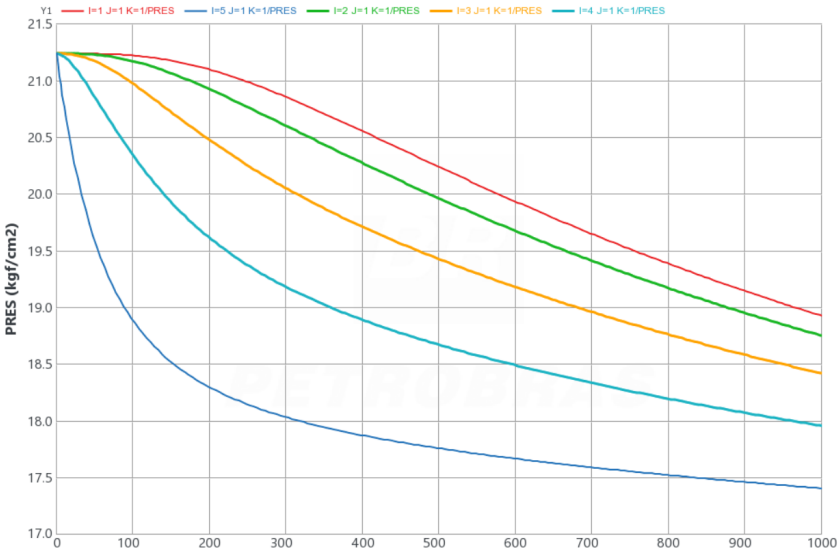

Figure 4.12: Pressure curve shown in each element during the simulation.

As the pressure in all elements decreases, the gas phase, non-existent at the start of the simulation, begins to appear. This event is shown by the S_g values increasing in Figure 4.13. In this sense, we can see that the cells, from right to left in Figure 4.11, begin to change from monophasic liquid state to a biphasic state. Naturally, the element in which the producer well is located is the first to present gas phase formation. The reason is it has the lowest value of pressure among the other cells, as shown in Figure 4.12. Also, as the

pressure decreases sequentially from the producer well element to the injector well cell, the S_o and S_g curves decreases and increases, respectively, at this same element order.

Finally, Figures 4.14 and 4.15 show the oil and gas molar fraction of the three components simulated. During the simulation, as the components migrate from the oil phase to form gas phase, y_i values start to increase.

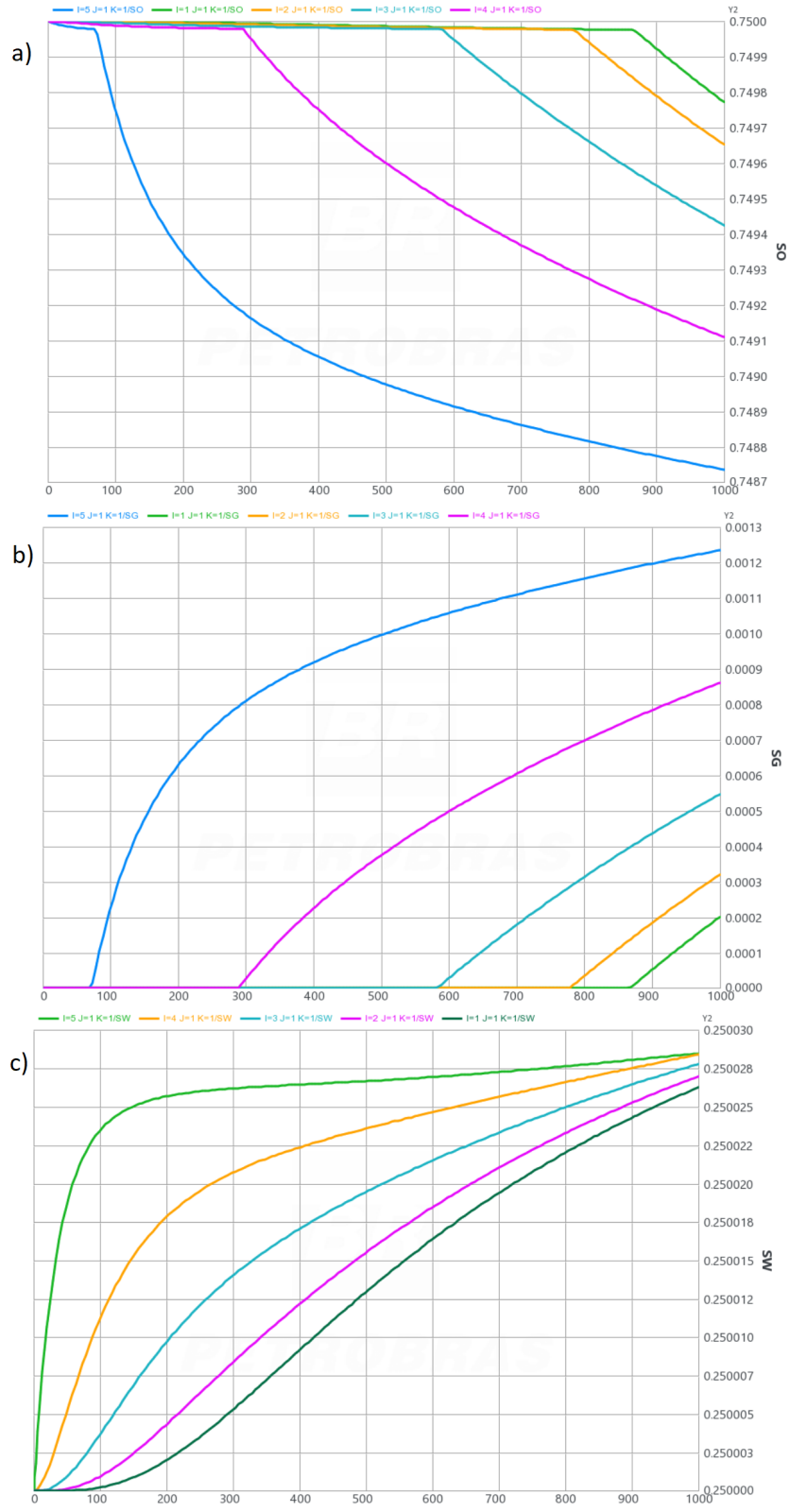

Figure 4.13: S_o, S_g and S_w curves shown in each element during the simulation.

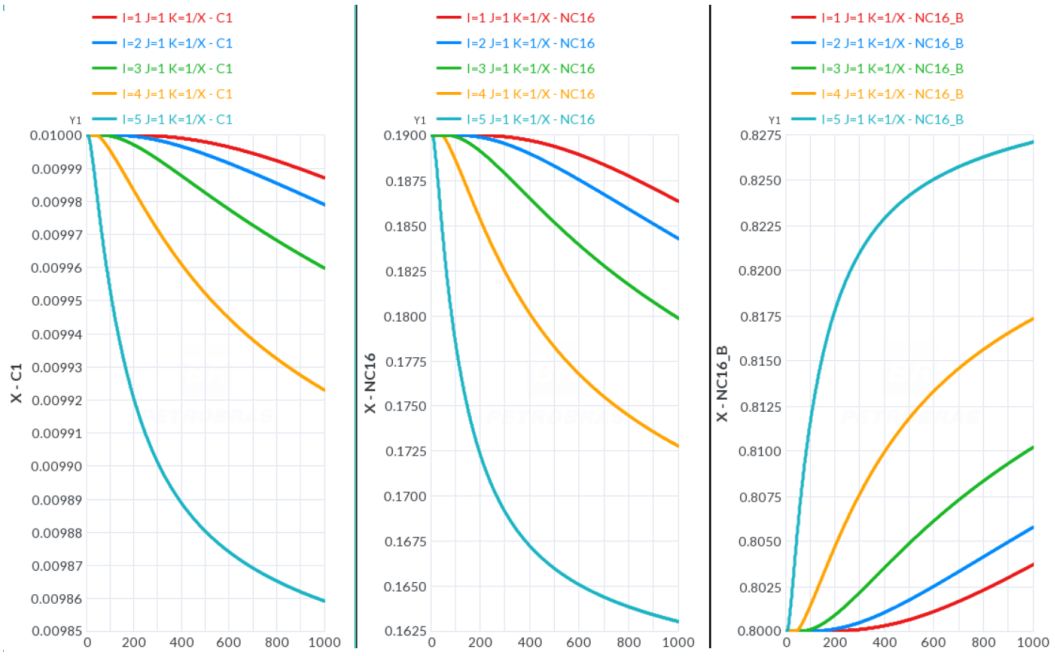

Figure 4.14: Oil molar fraction curves of the hydrocarbons shown in each element during the simulation.

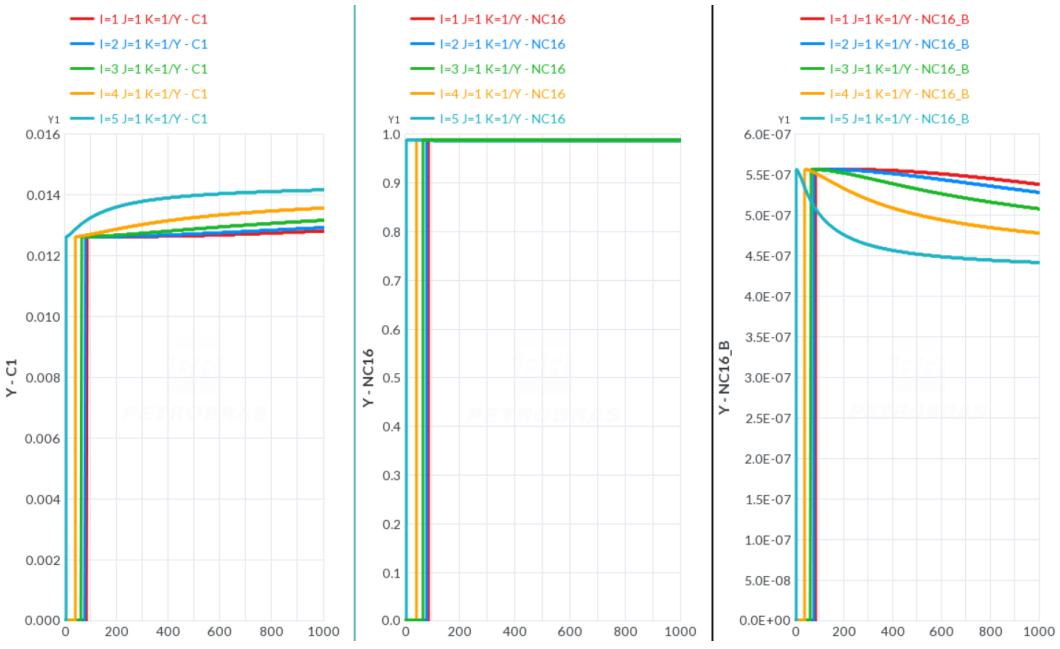

Figure 4.15: Gas molar fraction curves of the hydrocarbons shown in each element during the simulation.

5
Conclusions

This work aimed to add compositional simulation capabilities to the GSIM simulator, in addition to the already existing black oil modeling features. The new compositional formulation, based on the original work of Coats [68], was implemented in the simulator using a plugin architecture. The newly implemented model brings several advantages to GSIM, allowing it to provide more detailed and accurate production forecasts from petroleum reservoirs with compressible fluids that otherwise could not be addressed using a simple black oil approach. As a result, compositional changes, both in the porous media and the production stream, can be captured throughout the life of a petroleum field.

Additionally, the simulator provides a framework for modeling other physical and chemical phenomena where mass transfer between phases plays an import role, such as miscible oil recovery and water alternating gas (WAG) development strategies. Throughout the chapters, the theory involved in building a compositional model was revisited, including a detailed review of cubic equations of state (EOS), flash calculations and fugacities. In addition, the Topsim plugin architecture framework upon which GSIM is built was thoroughly discussed, indicating the straightforward features that easily enables GSIM to be extended by the addition of new functionalities.

Finally, the results of selected tests generated by GSIM were compared with those obtained from the CMG-GEM compositional reservoir simulator. Considering the similar results obtained from both simulators, it indicates that GSIM is adequate as a framework for reservoir simulation research, thus allowing for proof of concepts of innovative solutions required for tackling the challenging problems faced nowadays by the Oil & Gas Industry.

5.1
Suggestions for future work

The following research topics are suggested for future work:

– **Implementation of alternative grids:** As mentioned in Section 1.1.2, the GSIM simulator selected the Cartesian centered blocks method, a structured grid, in order to partition the simulated reservoir domain.

However, other alternative approaches within the structured grid group, such as the corner point, allow for a better representation of reservoirs highly faulted. The corner point method possibilities the match between the cell edges and the highly-faulted boundaries of the reservoir by considering arbitrary directions consideration for its cell edge construction. In fact, considering unstructured meshes can represent the reservoir domain even better than the corner point approach.

- **Implementation of a dual porosity/ permeability compositional formulations:** Most of the reservoirs simulated have some form of fracture, which makes it difficult to predict the fluid flow behavior accurately. A fractured porous medium is defined as a system of interconnected fractures through which the reservoir fluid flows. Within this specific porous medium, the porosity and permeability properties can vary abruptly between the fractures and the porous rock blocks (matrix blocks). Therefore, modeling the fluid behavior through this entire type of porous medium would require a high computational cost. In order to avoid these problems, two widely concepts are used: dual porosity (and single permeability) and dual porosity/permeability. They suggest dividing the porous medium into two continuum regions: the fractures and the matrix. Then, the fluid behavior is computed in the void space made by these two regions.
- **Implementation of alternatives formulations:** The compositional model involves solving $n_c(n_p - 1) + n_p + 1$ variables and equations. Each formulation proposes to solve this set by specifying which primary variables and equations are to be solved according to the degree of implicitness used. In this work, the Coats' formulation was used, which employs the FI method. Although FI is considered to be a stable method, it is computationally very expensive when working with large matrices. Other formulations, such as Adaptive Implicit Method (AIM), are able to present satisfactory results using a lower degree of implicitness and, therefore, consuming less computational cost.
- **Improvement of the rock-fluid interaction representation:** In reservoir simulations, several simplifications are made that provide moderate results for less computational effort. However, for certain cases of complex reservoirs, these assumptions can lead to inaccurate predictions. One possible solution is to improve the representation of the rock-fluid interaction, such as the hysteresis phenomena resulting from the fluid saturation history. Hysteresis is defined as the difference in the relationship between fluid saturations and relative permeability during the imbibition

and drainage processes. Silva et al. study [75] applies two widely used hysteresis models (Killough and Larsen and Skauge) in reservoir simulations and details how these models lead to a better prediction of oil recovery.

Bibliography

[1] FERNANDES, B. R. B.. **Implicit and semi-implicit techniques for the compositional petroleum reservoir simulation based on volume balance**. 2014.

[2] CHEN, Z.; GUANREN, H. ; YUANLE, M.. **Computational Methods for Multiphase Flows in Porous Media**. Ak Peters Series. Society for Industrial and Applied Mathematics, 2006.

[3] CAO, H.. **Development of techniques for general purpose simulators**. Stanford University, 2002.

[4] SANTOS, L. O. S. D.. **Development of a multi-formulation compositional simulator**. PhD thesis, 2013.

[5] ERTEKIN, T.; ABOU-KASSEM, J. H. ; KING, G. R.. **Basic applied reservoir simulation**, volume 7. Society of Petroleum Engineers Richardson, TX, 2001.

[6] DUARTE, L.. **TopSim: A plugin-based framework for large-scale numerical analysis**. PhD thesis, PhD thesis, Pontifical Catholic University of Rio de Janeiro, Rio de Janeiro . . . , 2016.

[7] AMOOIE, M. A.; MOORTGAT, J.. **Higher-order black-oil and compositional modeling of multiphase compressible flow in porous media**. International Journal of Multiphase Flow, 105, 2018.

[8] JACOBY, R.; BERRY JR, V.. **A method for predicting depletion performance of a reservoir producing volatile crude oil**. Transactions of the AIME, 210(01):27–33, 1957.

[9] DELSHAD, M.; THOMAS, S. G. ; WHEELER, M. F.. **Parallel numerical reservoir simulations of nonisothermal compositional flow and chemistry**. SPE Journal, 16(02):239–248, 2011.

[10] BRANTFERGER, K. M.; POPE, G. ; SEPEHRNOORI, K.. **Development of a thermodynamically consistent, fully implicit, equation-of-state, compositional steamflood simulator**. In: SPE RESERVOIR SIMULATION CONFERENCE, p. SPE–21253. SPE, 1991.

[11] AAVATSMARK, I.; BØE, Ø.; REISO, E. ; TEIGLAND, R.. **Mpfa for faults and local refinements with application to field simulations**. In: ECMOR VII-7TH EUROPEAN CONFERENCE ON THE MATHEMATICS OF OIL RECOVERY, p. cp–57. European Association of Geoscientists & Engineers, 2000.

[12] NACUL, E.; LEPRETRE, C.; PEDROSA JR, O.; GIRARD, P. ; AZIZ, K.. **Efficient use of domain decomposition and local grid refinement in reservoir simulation**. In: SPE ANNUAL TECHNICAL CONFERENCE AND EXHIBITION, p. SPE–20740. SPE, 1990.

[13] PEDROSA, O. A.; AZIZ, K.. **Use of a hybrid grid in reservoir simulation**. SPE Reservoir Engineering, 1(06):611–621, 1986.

[14] AZIZ, K.. **Petroleum reservoir simulation**. Applied Science Publishers, 476, 1979.

[15] AZIZ, K.. **Reservoir simulation grids: opportunities and problems**. Journal of Petroleum Technology, 45(07):658–663, 1993.

[16] FARMER, C. L.. **Geological modelling and reservoir simulation**. In: MATHEMATICAL METHODS AND MODELLING IN HYDROCARBON EXPLORATION AND PRODUCTION, p. 119–212. Springer, 2005.

[17] HEINEMANN, Z. E.. **Gridding techniques in reservoir simulation**. In: PROCEEDINGS OF 1ST AND 2ND INTERNATIONAL FORUM ON RESERVOIR SIMULATION, ALPBACH, AUSTRIA., 1988.

[18] HEINEMANN, Z.; BRAND, C.; MUNKA, M. ; CHEN, Y.. **Modeling reservoir geometry with irregular grids**. In: SPE RESERVOIR SIMULATION CONFERENCE, p. SPE–18412. SPE, 1989.

[19] FLANDRIN, N.; BOROUCHAKI, H. ; BENNIS, C.. **3d hybrid mesh generation for reservoir simulation**. International journal for numerical methods in engineering, 65(10):1639–1672, 2006.

[20] ISLAM, M. R.; ABOU-KASSEM, J. H.; HOSSAIN, M.; MOUSAVIZADEGAN, S. H. ; MUSTAFIZ, S.. **Advanced petroleum reservoir simulation: Towards developing reservoir emulators**. John Wiley & Sons, 2016.

[21] NARDEAN, S.; FERRONATO, M. ; ABUSHAIKHA, A.. **Linear solvers for reservoir simulation problems: An overview and recent developments**. Archives of Computational Methods in Engineering, 29(6):4341–4378, 2022.

[22] HURTADO, F. S. V.; MALISKA, C. R.; DA SILVA, A. F. C. ; CORDAZZO, J.. **A quadrilateral element-based finite-volume formulation for the simulation of complex reservoirs**. In: SPE LATIN AMERICA AND CARIBBEAN PETROLEUM ENGINEERING CONFERENCE, p. SPE–107444. SPE, 2007.

[23] EDWARDS, M. G.; ROGERS, C. F.. **A flux continuous scheme for the full tensor pressure equation**. In: ECMOR IV-4TH EUROPEAN CONFERENCE ON THE MATHEMATICS OF OIL RECOVERY, p. cp–233. European Association of Geoscientists & Engineers, 1994.

[24] EWING, R. E.; WHEELER, M. F. ; OTHERS. **The approximation of the pressure by a mixed method in the simulation of miscible displacement**. RAIRO. Analyse numérique, 17(1):17–33, 1983.

[25] DOUGLAS JR, J.; PEACEMAN, D. W. ; RACHFORD JR, H.. **A method for calculating multi-dimensional immiscible displacement**. Transactions of the AIME, 216(01):297–308, 1959.

[26] AU, A. D.; BEHIE, G.; RUBIN, B. ; VINSOME, P.. **Techniques for fully implicit reservoir simulation**. In: SPE ANNUAL TECHNICAL CONFERENCE AND EXHIBITION, p. SPE–9302. SPE, 1980.

[27] SHELDON, J.; CARDWELL, W. ; OTHERS. **One-dimensional, incompressible, noncapillary**. Two-phase fluid flow in a porous medium, 1959.

[28] THOMAS, G. W.; THURNAU, D. H.. **Reservoir simulation using an adaptive implicit method**. Society of Petroleum Engineers Journal, 23(05):759–768, 1983.

[29] ESLER, K.; GANDHAM, R.; PATACCHINI, L.; GARIPOV, T.; SAMARDZIC, A.; PANFILI, P.; CARESANI, F.; PIZZOLATO, A. ; COMINELLI, A.. **A graphics processing unit–based, industrial grade compositional reservoir simulator**. SPE Journal, 27(01):597–612, 2022.

[30] FORSYTH JR, P.; SAMMON, P.. **Practical considerations for adaptive implicit methods in reservoir simulation**. Journal of Computational Physics, 62(2):265–281, 1986.

[31] JENNY, P.; LEE, S. H. ; TCHELEPI, H. A.. **Adaptive fully implicit multi-scale finite-volume method for multi-phase flow and transport in heterogeneous porous media**. Journal of Computational Physics, 217(2):627–641, 2006.

[32] MONCORGÉ, A.; TCHELEPI, H. A. ; JENNY, P.. **Modified sequential fully implicit scheme for compositional flow simulation**. Journal of Computational Physics, 337:98–115, 2017.

[33] MONCORGÉ, A.; TCHELEPI, H. A. ; JENNY, P.. **Sequential fully implicit formulation for compositional simulation using natural variables**. Journal of Computational Physics, 371:690–711, 2018.

[34] JIANG, J.; TCHELEPI, H. A.. **Nonlinear acceleration of sequential fully implicit (sfi) method for coupled flow and transport in porous media**. Computer Methods in Applied Mechanics and Engineering, 352:246–275, 2019.

[35] MØYNER, O.; MONCORGÉ, A.. **Nonlinear domain decomposition scheme for sequential fully implicit formulation of compositional multiphase flow**. Computational Geosciences, 24(2):789–806, 2020.

[36] LI, J.; TOMIN, P. ; TCHELEPI, H.. **Sequential fully implicit newton method for compositional flow and transport**. Journal of Computational Physics, 444:110541, 2021.

[37] JIANG, J.; TOMIN, P. ; ZHOU, Y.. **Inexact methods for sequential fully implicit (sfi) reservoir simulation**. Computational Geosciences, 25:1709–1730, 2021.

[38] MAGRAS, J.; QUANDALLE, P. ; BIA, P.. **High-performance reservoir simulation with parallel athos**. In: SPE RESERVOIR SIMULATION CONFERENCE, p. SPE–66342. SPE, 2001.

[39] HU, X.; WU, S.; WU, X.-H.; XU, J.; ZHANG, C.-S.; ZHANG, S. ; ZIKATANOV, L.. **Combined preconditioning with applications in reservoir simulation**. Multiscale Modeling & Simulation, 11(2):507–521, 2013.

[40] WANG, K.; LIU, H.; LUO, J. ; CHEN, Z.. **Efficient cpr-type preconditioner and its adaptive strategies for large-scale parallel reservoir simulations**. Journal of Computational and Applied Mathematics, 328:443–468, 2018.

[41] VINSOME, P. K.. **Orthomin, an iterative method for solving sparse sets of simultaneous linear equations**. In: SPE SYMPOSIUM ON NUMERICAL SIMULATION OF RESERVOIR PERFORMANCE, p. SPE–5729. SPE, 1976.

[42] SAAD, Y.; SCHULTZ, M. H.. **Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems**. SIAM Journal on scientific and statistical computing, 7(3):856–869, 1986.

[43] VAN DER VORST, H. A.. **Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems**. SIAM Journal on scientific and Statistical Computing, 13(2):631–644, 1992.

[44] BENZI, M.. **Preconditioning techniques for large linear systems: a survey**. Journal of computational Physics, 182(2):418–477, 2002.

[45] BEHIE, A.; VINSOME, P.. **Block iterative methods for fully implicit reservoir simulation**. Society of Petroleum Engineers Journal, 22(05):658–668, 1982.

[46] WALLIS, J. R.; KENDALL, R. P. ; LITTLE, T.. **Constrained residual acceleration of conjugate residual methods**. In: SPE RESERVOIR SIMULATION CONFERENCE, p. SPE–13536. SPE, 1985.

[47] BROWN, G. L.; COLLINS, D. A. ; CHEN, Z.. **Efficient preconditioning for algebraic multigrid and red-black ordering in adaptive-implicit black-oil simulations**. In: SPE RESERVOIR SIMULATION CONFERENCE, p. D011S002R006. SPE, 2015.

[48] BASTOS, T. S.. **Development of a multipurpose reservoir simulator based on a plugin architecture**. 2021.

[49] COREY, A. T.. **Mechanics of immiscible fluids in porous media**. Water Resources Publication, 1994.

[50] STONE, H.. **Estimation of three-phase relative permeability and residual oil data**. Journal of Canadian Petroleum Technology, 12(04), 1973.

[51] RACHFORD JR, H. H.; RICE, J.. **Procedure for use of electronic digital computers in calculating flash vaporization hydrocarbon equilibrium**. Journal of Petroleum Technology, 4(10):19–3, 1952.

[52] PENG, D.-Y.; ROBINSON, D. B.. **A new two-constant equation of state**. Industrial & Engineering Chemistry Fundamentals, 15(1):59–64, 1976.

[53] COUSSY, O.. **Poromechanics**. Wiley, Chichester, England ; Hoboken, NJ, 2nd ed edition, 2004.

[54] JOSSI, J. A.; STIEL, L. I. ; THODOS, G.. **The viscosity of pure substances in the dense gaseous and liquid phases**. AIChE Journal, 8(1):59–63, 1962.

[55] GONZALEZ ABAD, K. G.. **Development of a compositional reservoir simulator for asphaltene precipitation based on a thermodynamically consistent model**. PhD thesis, 2013.

[56] COREY, A. T.; RATHJENS, C.; HENDERSON, J. ; WYLLIE, M.. **Three-phase relative permeability**. Journal of Petroleum Technology, 8(11):63–65, 1956.

[57] SNELL, R.. **Three-phase relative permeability in an unconsolidated sand**. J. Inst. Pet, 48(459):80–88, 1962.

[58] PEACEMAN, D. W.. **Fundamentals of numerical reservoir simulation**. Elsevier, 2000.

[59] WHITSON, C. H.; BRULE, M. R.. **Phase behavior**, volume 20. Henry L. Doherty Memorial Fund of AIME, Society of Petroleum Engineers . . . , 2007.

[60] BEAR, J.. **Dynamics of fluids in porous media**. Courier Corporation, 2013.

[61] MICHELSEN, M. L.. **The isothermal flash problem. part i. stability**. Fluid phase equilibria, 9(1):1–19, 1982.

[62] MICHELSEN, M. L.. **The isothermal flash problem. part ii. phase-split calculation**. Fluid phase equilibria, 9(1):21–40, 1982.

[63] WILSON, G. M.. **A modified redlich-kwong equation of state, application to general physical data calculations**. In: 65TH NATIONAL AICHE MEETING, CLEVELAND, OH, volume 15, 1969.

[64] NGHIEM, L. X.; AZIZ, K. ; LI, Y.. **A robust iterative method for flash calculations using the soave-redlich-kwong or the peng-robinson equation of state**. Society of Petroleum Engineers Journal, 23(03):521–530, 1983.

[65] PEACEMAN, D. W.. **Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988)**. Society of Petroleum Engineers Journal, 18(03):183–194, 1978.

[66] CAO, H.; AZIZ, K.. **Performance of impsat and impsat-aim models in compositional simulation**. In: SPE ANNUAL TECHNICAL CONFERENCE AND EXHIBITION, p. SPE–77720. SPE, 2002.

[67] COLLINS, D.; NGHIEM, L.; LI, Y.-K. ; GRABONSTOTTER, J.. **An efficient approach to adaptive-implicit compositional simulation with an equation of state**. SPE reservoir engineering, 7(02):259–264, 1992.

[68] COATS, K. H.. **An equation of state compositional model**. Society of Petroleum Engineers Journal, 20(05):363–376, 1980.

[69] AZIZ, K.; WONG, T.. **Considerations in the development of multipurpose reservoir simulation models, proceedings of the 1st and 2nd international forum on reservoir simulation**. Alpbach, Austria, September, p. 12–16, 1988.

[70] ABOU-KASSEM, J. H.; OSMAN, M. E. ; ZAID, A. M.. **Architecture of a multipurpose simulator**. Journal of Petroleum Science and Engineering, 16(4):221–235, 1996.

[71] STRÎMBEI, C.; DOSPINESCU, O.; STRAINU, R.-M. ; NISTOR, A.. **Software architectures-present and visions**. Informatica Economica, 19(4):13, 2015.

[72] CELES, W.; PAULINO, G. H. ; ESPINHA, R.. **A compact adjacency-based topological data structure for finite element mesh representation**. International journal for numerical methods in engineering, 64(11):1529–1556, 2005.

[73] MANSOOR, U.; KESSENTINI, M.; MAXIM, B. R. ; DEB, K.. **Multi-objective code-smells detection using good and bad design examples**. Software Quality Journal, 25:529–552, 2017.

[74] SCHENK, O.; GÄRTNER, K.. **Solving unsymmetric sparse systems of linear equations with pardiso**. Future Generation Computer Systems, 20(3):475–487, 2004.

[75] DA SILVA, D. A. R.; HERNANDEZ, J. A. M. ; BARILLAS, J. L. M.. **Relative permeability hysteresis analysis in a reservoir with characteristics of the brazilian pre-salt**. Research, Society and Development, 12(2):e24712239842–e24712239842, 2023.