UNIVERSIDADE FEDERAL DE SANTA CATARINA

CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

FELIPE MIGUEL WALLNER GIACOMELLI

A SPEEDUP COMPARISON BETWEEN PARALLEL IMPLEMENTATIONS OF GEOMECHANICAL COUPLING SCHEMES

Florianópolis

2019

UNIVERSIDADE FEDERAL DE SANTA CATARINA

CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

FELIPE MIGUEL WALLNER GIACOMELLI

A SPEEDUP COMPARISON BETWEEN PARALLEL IMPLEMENTATIONS OF GEOMECHANICAL COUPLING SCHEMES

Trabalho de Conclusão do Curso de graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina, como requisito parcial para a obtenção do Título de Bacharel em Engenharia Mecânica.

Orientador: António Fábio Carvalho da Silva, Dr. Eng.
Coorientador: Hermínio Tasinafo Honório, Dr. Eng.

Florianópolis

2019

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Giacomelli, Felipe Miguel Wallner
A Speedup Comparison Between Parallel Implementations of Geomechanical Coupling Schemes / Felipe Miguel Wallner Giacomelli ; orientador, António Fábio Carvalho da Silva, coorientador, Hermínio Tasinafo Honório, 2019.
72 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Mecânica, Florianópolis, 2019.

Inclui referências.

1. Engenharia Mecânica. 2. Poroelasticity. 3. Element based Finite Volume Method. 4. Domain Decomposition Method. 5. Parallel computing. I. da Silva, António Fábio Carvalho. II. Honório, Hermínio Tasinafo. III. Universidade Federal de Santa Catarina. Graduação em Engenharia Mecânica. IV. Título.

FELIPE MIGUEL WALLNER GIACOMELLI

A SPEEDUP COMPARISON BETWEEN PARALLEL IMPLEMENTATIONS OF GEOMECHANICAL COUPLING SCHEMES

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção do Título de Engenheiro Mecânico e aprovado em sua forma final pela Comissão Examinadora e pelo Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina.

Florianópolis, 17 de dezembro de 2019.

Prof. Carlos Enrique Niño Bohórquez, Dr. Eng.
Coordenador do Curso

Comissão Examinadora:

Prof. António Fábio Carvalho da Silva, Dr. Eng
Orientador
Universidade Federal de Santa Catarina

Hermínio Tasinafo Honório, Dr. Eng
Coorientador
Universidade Federal de Santa Catarina

Prof. Eduardo Alberto Fancello, Dr. Eng.
Universidade Federal de Santa Catarina

Prof. Carlos Enrique Niño Bohórquez, Dr. Eng.
Universidade Federal de Santa Catarina

Às minhas tias Sara e Jussara,
aos meus pais Ivan e Silvia.

Acknowledgements

This work has been partially supported by *Petrobras* through project number 5850.010924.18.9, entitled "Malhas Tridimensionais Híbridas para a Solução do Problema Acoplado Fluxo-Geomecânica", within *Rede de Geomecânica.*

Resumo

Neste trabalho é feita uma comparação do desempenho em paralelo entre dois métodos de acoplamento para poroelasticidade. No *Fully Implicit Method*, o acoplamento escoamento-geomecânica é implicitamente realizado na solução do sistema linear, uma vez que as equações de balanço mecânico e de conservação da massa são resolvidas num único sistema. No que lhe concerne, o *Fixed-Stress Split* consiste em resolver estas duas equações separadamente, e o acoplamento se dá através de um processo iterativo que é repetido a cada passo no tempo. Ambos os métodos foram paralelizados através do *Método de Decomposição do Domínio*, onde o domínio de cálculo é dividido em subdomínios, e cada um destes é atribuído a um processador. Como os processadores trabalham simultaneamente, o tempo necessário para determinar as incógnitas em todo o domínio equivale ao tempo gasto por um processador para definir as incógnitas no seu subdomínio somado com o tempo de comunicação entre processadores. O teste de desempenho é realizado através da solução de dois problemas de poroelasticidade: a coluna de Terzaghi e a esfera de Cryer. Embora o Fixed-Stress Split tenha apresentado o maior aumento de desempenho em decorrência da paralelização ele gastou mais tempo de CPU que o Fully Implicit Method em todas as simulações realizadas.

Palavras-chave: Engenharia Mecânica. Geomecânica. Método dos Volumes Finitos baseado em Elementos. Método de Decomposição do Domínio. Computação em paralelo.

Abstract

In the present work, a speedup comparison between two poroelasticity coupling schemes is performed. In the *Fully Implicit Method*, the geomechanics-flow coupling is implicitly accomplished in the solution of the linear system, since the mechanical balance and mass conservation equations are solved in a single linear system. In turn, the *Fixed-Stress Split* consists of segregately solving these equations, and the coupling is achieved through an iterative procedure at each time step. Both methods have been parallelized by the *Domain Decomposition Method*, in which the computational domain is divided into subdomains, which are then assigned to a processor. Once that all processors work simultaneously, the time required to evaluate the unknowns throughout the domain is equivalent to the time taken by a processor to elicit the unknowns concerning its subdomain in addition to the inter-processor communication time. The performance test is carried out through the solution of two poroelasticity problems: Terzaghi's column and Cryer's sphere. Although the Fixed-Stress Split has shown a larger speedup, it is still outperformed by the Fully Implicit Method in all simulated scenarios.

Keywords: Mechanical Engineering. Geomechanics. Element based Finite Volume Method. Domain Decomposition Method. Parallel computing.

List of figures

List of tables

List of abbreviations and acronyms

CFD	Computational Fluid Dynamics
CGNS	CFD General Notation System
DDM	Domain Decomposition Method
DEI	DivideEtImpera
EbFVM	Element based Finite Volume Method
FIM	Fully Implicit Method
FSS	Fixed-Stress Split
I/O	Input/Output
PETSc	Portable, Extensible Toolkit for Scientific Computation

List of symbols

Latin Letters

$\mathbb{A}$	Global accumulation matrix
$\boldsymbol{b}$	Independent term vector
$\boldsymbol{B}$	Gradient operator
$\boldsymbol{B}_s$	Symmetric gradient operator
c_b	Bulk compressibility
c_f	Fluid compressibility
c_s	Solid compressibility
c_v	Consolidation coefficient
$\mathbb{C}$	Fourth-order constitutive tensor
$\boldsymbol{D}$	Matrix containing the shape function local derivatives
$\mathbb{D}$	Fixed stress
G	1st Lamè's parameter (shear modulus)
$\boldsymbol{g}$	Acceleration gravity vector
$\boldsymbol{G}_{ip}$	Darcy's gravity matrix
$\mathbb{G}$	Global Darcy's gravity matrix
h_i	Difference in hydrostatic pressure
$\boldsymbol{H}_{ip}$	Darcy's velocity matrix
$\mathbb{H}$	Global Darcy's velocity matrix
$\boldsymbol{I}$	Second-order identity tensor
j	Iterative level
$\boldsymbol{J}$	Jacobian matrix
K	Bulk modulus
$\boldsymbol{k}$	Absolute permeability tensor
k	Absolute permeability
$\boldsymbol{K}_{ip}$	Stiffness matrix
$\mathbb{K}$	Global stiffness matrix
l	Characteristic length
$\boldsymbol{L}_{ip}$	Pore pressure matrix
$\mathbb{L}$	Global sore pressure matrix
m	Number of vertices
m_v	Confined compressibility
$\hat{\boldsymbol{n}}$	Unitary normal vector

N_v	Amount of control volumes of the domain
N_t	Amount of time steps simulated
N_i	Element shape function associated to the ith vertex
$\boldsymbol{N}$	Array containing the element shape functions
$\boldsymbol{N}_s$	Special arrangement of the element shape functions
p	Pore pressure
$\bar{p}$	Prescribed pore pressure
p_w	Well top pressure
$\boldsymbol{P}$	Pore pressure of all grid vertices
q	Rate of fluid volume injected or removed per unit of volume
q_t	Total volumetric flow across the wall
$\frac{1}{Q}$	Biot's modulus
$\boldsymbol{Q}_{ip}$	Volumetric strain matrix
$\mathbb{Q}$	Global volumetric strain matrix
r_{eq}	Equivalent radius
r_w	Wellbore radius
$\boldsymbol{r}$	Position vector
$\dot{s}$	Subelement
$\boldsymbol{s}$	Face area vector
$\boldsymbol{s}_s$	Symmetric face area vector
$\boldsymbol{t}$	Surface traction vector
t_d	Dimensionless time
u	Displacement component in x direction
$\boldsymbol{u}$	Displacement vector
$\bar{\boldsymbol{u}}$	Prescribed displacement vector
$\boldsymbol{U}$	Displacements of all grid vertices
v	Displacement component in y direction
$\boldsymbol{v}$	Darcy's velocity
w	Displacement component in z direction
WI	Well index
x	Coordinate x in the real domain
y	Coordinate y in the real domain
z	Coordinate z in the real domain
z_{top}	Coordinate z of the well top

Greek Letters

α	Biot's coefficient
γ_f	Fluid specific weight
Γ	Surface boundary
ε	Strain
ε_v	Volumetric strain
$\boldsymbol{\varepsilon}$	Strain tensor
θ	Variation in water content
λ	2^{st} Lamè's parameter
Λ	Set of control volumes intercepted by a well
μ	Fluid viscosity
ν	Poisson's ratio
ρ	Porous medium average density
ρ_f	Fluid phase density
ρ_s	Solid phase density
σ	Prescribed load
σ_v	Total volumetric stress
σ'_v	Effective volumetric stress
$\boldsymbol{\sigma}$	Total stress tensor
$\boldsymbol{\sigma}'$	Effective stress tensor
τ	Coupling intensity
ϕ	Porosity
Φ	General scalar quantity
$\boldsymbol{\Phi}$	Vector containing the general scalar quantities associated to the element vertices
$\dot{\omega}$	Volumetric flux
Ω	Domain of calculus

Special symbols

∂	Partial derivative
$d\Omega$	Differential volume in the real domain
$d\Gamma$	Differential area
dt	Differential time step
Δt	Time step size
Δx	Length of the control volume

Δy	Width of the control volume
Δz	Height of the control volume
$\Delta\Omega_{\dot{s}}$	Volume of the sub-element
$\mathscr{H}$	Set of sub-elements
∇	Nabla operator
∇_s	Symmetric nabla operator

Superscripts

p	Relative to pore pressure
$\boldsymbol{u}$	Relative to displacement
o	Evaluated in the previous time step

Subscripts

0	Initial state
D	Dirichlet boundary condition
e	Relative to element e
ip	Relative to integration point ip
N	Neumann boundary condition
x, y, z	Derivative relative to global coordinates
ξ, η, γ	Derivative relative to local coordinates

List of definitions

Contents

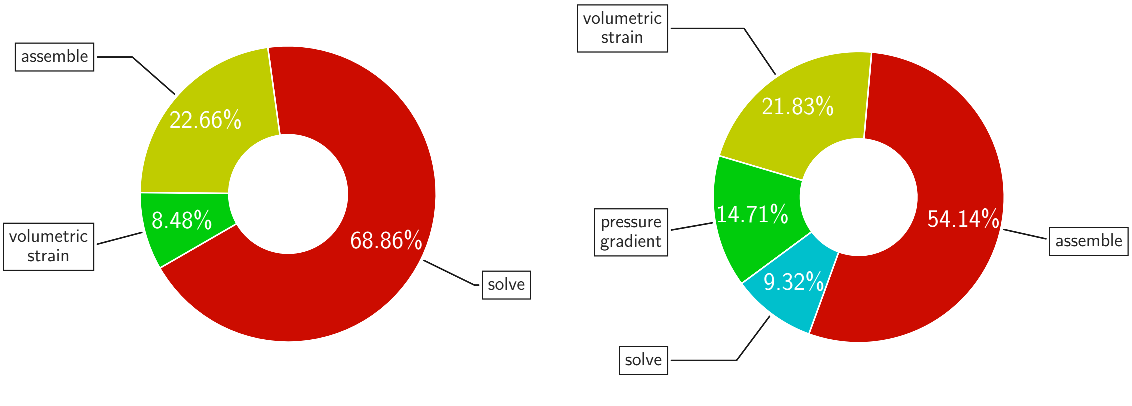
1 Introduction

The mechanical behaviour of a deformable porous medium, whose pores are saturated with fluid, is of importance in several engineering applications, ranging from biomechanics (FANCELLO, 2019) to the oil and gas exploitation (MURAD, 2019). For instance, the petroleum flow through the reservoir pores causes a pressure gradient, promoting a force unbalance in the geological formation. To re-establish the mechanical balance of the solid-fluid system, the reservoir deforms, affecting the petroleum flow. Thus, two coupled *phenomena* must be considered in the determination of the porous medium behaviour: fluid flow and geomechanics.

With the purpose of modelling the coupling between these two *phenomena*, it is common the usage of Biot's consolidation model (BIOT, 1941), which has become the cornerstone of the poroelasticity theory. Despite that this mathematical model is well established, various approaches regarding the numerical coupling of its equations are presented in the scientific community.

Among these techniques, there are the *Fully Implicit Method* and the *Fixed-Stress Split* (KIM; TCHELEPI; JUANES, 2011). The former couples the model equations by solving them in a single linear system, therefore no iterative routine is required. Although, this seems convenient, the Fully Implicit Method may yield very ill-conditioned linear systems (GAMBOLATI; FERRONATO; JANNA, 2011), besides providing huge linear systems as fine grids are utilised. On the other hand, the Fixed-Stress Split iteratively couples the geomechanical and fluid flow models. Hence, two linear system must be solved at each iteration, which may also be expensive, time wise. The most time-consuming tasks concerning each of these schemes are pictured in Fig. 1.1.

Figure 1.1 – Most time-consuming tasks for two coupling schemes.



(a) Fully Implicit Method

(b) Fixed-Stress Split

Source: Own authorship.

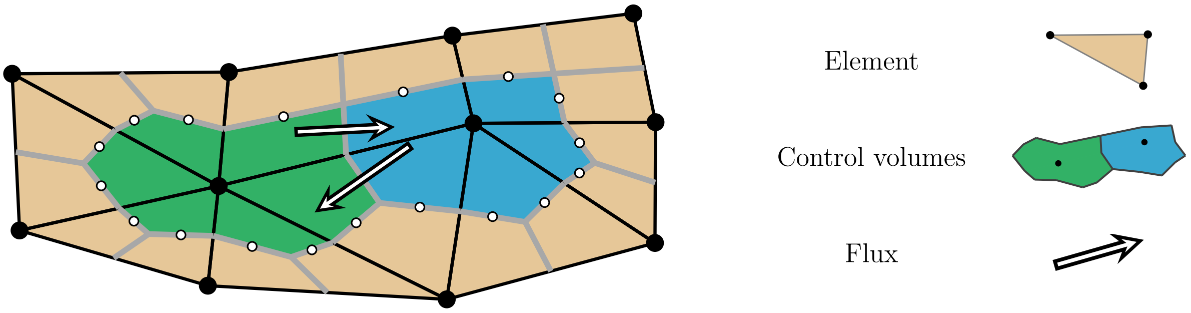
1.1 The task

In order to verify which coupling scheme is faster, a performance comparison between FIM and FSS have already been performed (HONÓRIO et al., 2019b). As a result, it was determined that the former outperforms the latter for all simulated scenarios of poroelasticity coupled with single-phase flow. However, no parallelization technique was employed in this study, leaving the parallel performance of both coupling schemes unknown.

Therefore, the purpose of this work is to elicit the parallel behaviour of the Fully Implicit Method and the Fixed-Stress Split in the solution of coupled geomechanics problems. This shall be achieved through the parallelization of in-house C++ library named *EFVLib2018*, which has several algorithms concerning the simulation of these scenarios.

Furthermore, this library utilises the *Element based Finite Volume Method* to numerically solve the partial differential equations that model the coupled *phenomena* (HONÓRIO et al., 2018). Being a control volume method, as shown in Fig. 1.2, the EbFVM guarantees the conservation of all unknown physical quantities for every control volume of the domain. Also, it is effortlessly applied to unstructured grids, thus providing substantial geometrical flexibility for the discretization of complex domains.

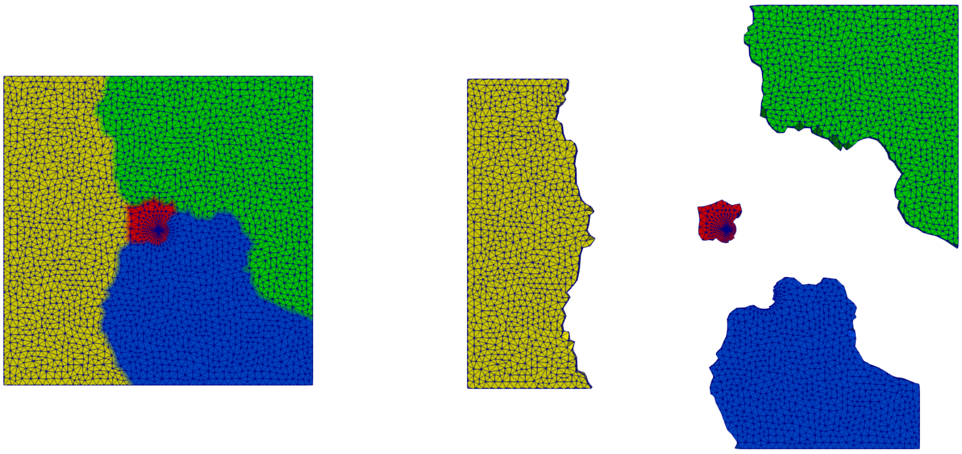
Figure 1.2 – Flux exchange between control volumes.



Source: Own authorship.

In turn, the parallelization technique chosen to fulfil this goal is the *Domain Decomposition Method*, as done by Grein (GREIN, 2015), given that it allows an even distribution of most simulation tasks among processors. This approach consists of dividing the computational domain into subdomains, as illustrated in Fig. 1.3, which are then distributed among processors. Thus, each processor works only in its subdomain, being the latter much smaller than the original domain. Since they all work simultaneously, the time spent by a single processor to perform a small scale simulation, adding on the inter-processor communication time, is equivalent to the time required by all processors to complete a large simulation case.

Figure 1.3 – Decomposition of a domain.



(a) Subdomains. (b) Exploded view.

Source: Own authorship.

1.2 Outline of the study

After introducing the subject of this study and defining the objectives that shall be achieved, the manner in which this work is organized is presented next.

In Chap. 2, the geomechanical and single-phase flow model equations are obtained starting from Cauchy's balance equation and a mass balance, respectively. Then, the initial and boundary conditions are displayed.

Before discussing the numerical formulation, the geometrical entities of an unstructured grid are introduced in Chap. 3. Also, it is elicited how these entities are employed in the computation of areas and volumes, as well as in the interpolation of scalar fields and the approximation of gradients.

Next, in Chap. 4, the mathematical model equations are discretized by the EbFVM. It is important to remark that this procedure requires the geometrical entities described in Chap. 3.

The numerical coupling of the discrete equations are exhibited in Chap. 5. Thus, a brief discussion in respect to the FIM and FSS is performed.

In what follows, Chap. 6 presents the DDM, eliciting the steps required for decomposing a computational grid, together with the open-source libraries supported its implementation.

After validating the coupled geomechanics algorithms of EFVLib2018, in Chap. 7, the comparison methodology is explained in detail. Then, the speedup achieved by both coupling schemes in two classical poroelasticity problems is shown, followed by a reservoir depletion simulation — closing this chapter.

At last, in Chap. 8 the final conclusions of the present work are stated, alongside suggestions for future studies.

2 Mathematical model

In this chapter, the mathematical model for poroelasticity coupled with single-phase flow is exhibited. Both geomechanical and flow formulations here introduced account for the response of the soil as a whole, *i.e*, there is no distinction between the individual contribution of its solid and fluid components. However, to express the system's response in terms of its constituents is quite useful, especially in the determination of the Biot's constants, which physically characterize the soil. Thus, to shed light on this characterization, a micromechanical analysis is performed after the presentation of the geomechanical and single-phase flows models. The former is introduced through the elicitation of the stress-strain and strain-displacement relations, as the latter is shown by a mass balance.

2.1 Geomechanical model

The starting point of the geomechanical model is the mechanical balance equations, which may be obtained through a balance of forces acting on the porous medium. Considering that its acceleration is zero, Cauchy's balance equation is achieved

$$\nabla \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} = 0, \tag{2.1}$$

in which $\boldsymbol{\sigma}$ is the second order stress tensor, $\boldsymbol{g}$ is the gravity vector, and ρ is the porous medium average density, defined by

$$\rho = \phi \rho_f + (1 - \phi) \rho_s, \tag{2.2}$$

where ρ_f, ρ_s represents the fluid and solid densities, respectively, and ϕ stands for porosity.

2.1.1 Terzaghi's effective stress

According to Terzaghi's effective stress principle (TERZAGHI, 1923), the total stress tensor $\boldsymbol{\sigma}$ acting on a porous material is partially supported by the solid structure and partially by the fluid phase. If the soil is assumed to be isotropic, the pore pressure cannot cause any shearing strain, such that

$$\boldsymbol{\sigma} - \boldsymbol{\sigma}_0 = \boldsymbol{\sigma'} - \alpha (p - p_0) \boldsymbol{I}, \tag{2.3}$$

where $\boldsymbol{I}$ is the identity second-order tensor, $\boldsymbol{\sigma'}$ is the effective stress acting on the solid phase, α is a physical constant called Biot's coefficient, and p is the pore pressure — the subscript "$_0$" indicates initial state.

The substitution of Eq. 2.3 in Eq. 2.1 leads to

$$\nabla \cdot (\boldsymbol{\sigma'} - \alpha(p - p_0)\boldsymbol{I} + \boldsymbol{\sigma}_0) + \rho \boldsymbol{g} = 0, \tag{2.4}$$

which may be rewritten as

$$\nabla \cdot (\boldsymbol{\sigma'} - \alpha p \boldsymbol{I}) + \rho \boldsymbol{g} = -\nabla \cdot (\boldsymbol{\sigma}_0 + \alpha p_0 \boldsymbol{I}). \tag{2.5}$$

2.1.2 Constitutive equation

The constitutive equation models the relation between stress and strain for a given material. In the present work, this relation is assumed to be linear and reversible under final equilibrium conditions, as in the generalized Hooke's law

$$\boldsymbol{\sigma}' = \mathbb{C} : \boldsymbol{\varepsilon}, \tag{2.6}$$

where $\mathbb{C}$ is a fourth-order constitutive tensor and $\boldsymbol{\varepsilon}$ is the second-order strain tensor. As the assumption of small strains is made, the relation between strain and displacement may be expressed by

$$\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T], \tag{2.7}$$

with $\boldsymbol{u}$ being the displacement vector.

Thus, given the symmetries of the stress ($\boldsymbol{\sigma}_{ij} = \boldsymbol{\sigma}_{ji}$) and strain ($\boldsymbol{\varepsilon}_{ij} = \boldsymbol{\varepsilon}_{ji}$) tensors, it is possible to infer the following symmetries of $\mathbb{C}$

$$\mathbb{C}_{ijkl} = \mathbb{C}_{jikl} = \mathbb{C}_{ijlk} = \mathbb{C}_{klij}. \tag{2.8}$$

Therefore, the number of independent constants required to represent $\mathbb{C}$ diminishes from 81 to 21. Moreover, the soil is considered to be isotropic linear elastic, which further reduces the amount of independent constants from 21 to 2. Hence, the tensor $\mathbb{C}$, may be defined as

$$\mathbb{C} = G(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda \delta_{ij}\delta_{kl}, \tag{2.9}$$

with G being the shear modulus and λ denoting the first Lamè's parameter,

$$\lambda = \frac{2G\nu}{1-2\nu}, \tag{2.10}$$

in which ν is the Poisson's ratio. Again, it is important to recall that G, ν and λ are properties regarding the solid-fluid system.

2.1.3 Voigt notation

If the symmetries discussed in Sec. 2.1.2 are taken into account, a more compact notation, known as the Voigt notation, may be put to use. Thus, Eq. 2.6 may be rewritten as

$$\begin{bmatrix} \sigma'_{11} \\ \sigma'_{22} \\ \sigma'_{33} \\ \sigma'_{23} \\ \sigma'_{13} \\ \sigma'_{12} \end{bmatrix} = \begin{bmatrix} 2G+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2G+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2G+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix}. \tag{2.11}$$

In order to write Eq. 2.5 employing the Voigt notation, it is necessary to define the symmetric nabla operator as

$$\nabla_s = \begin{bmatrix} \partial_x & 0 & 0 & 0 & \partial_z & \partial_y \\ 0 & \partial_y & 0 & \partial_z & 0 & \partial_x \\ 0 & 0 & \partial_z & \partial_y & \partial_x & 0 \end{bmatrix}^T. \tag{2.12}$$

Then, given that $\boldsymbol{\varepsilon} = \nabla_s \boldsymbol{u}$, it is possible to rewrite Eq. 2.5 as

$$\nabla_s^T(\mathbb{C}\nabla_s \boldsymbol{u} - \alpha p \boldsymbol{I}) \; = \; -\nabla_s^T(\boldsymbol{\sigma}_0 + \alpha p_0 \boldsymbol{I}) - \rho \boldsymbol{g}, \tag{2.13}$$

At last, it is important to recall the Voigt representation of the second-order tensor identity $\boldsymbol{I}$,

$$\boldsymbol{I} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T, \tag{2.14}$$

which shall be used throughout this work.

2.2 Flow model

The basis of the flow model is the mass conservation equation, which may be achieved through a mass balance performed in an elementary control volume. However, a new physical quantity shall first be defined, as done by Biot.

2.2.1 Biot's consolidation theory

According to Biot's consolidation theory (BIOT, 1941), the variation of water volume per unit volume of soil is measured by the *variation in water content*. This quantity may be defined as a linear relation between the *volumetric strain* ε_v and p, as in

$$\theta = \alpha\varepsilon_v + \frac{p}{Q}, \tag{2.15}$$

where θ is the variation in water content, and Q contains the Biot's modulus. Moreover, the considered strain is only volumetric given the assumed isotropy of the soil, thus shear strains cannot cause variations in the water content.

Furthermore, the fluid is considered to flow across the solid matrix pores according Darcy's law

$$\boldsymbol{v} = -\frac{\boldsymbol{k}}{\mu}(\nabla p - \rho_f \boldsymbol{g}), \tag{2.16}$$

with $\boldsymbol{v}$ being the fluid phase velocity relative to the solid skeleton, $\boldsymbol{k}$ is the absolute permeability tensor, and μ is the fluid viscosity.

Hence, having defined the variation in water content θ and the relative velocity $\boldsymbol{v}$, a fluid mass balance in an elementary control volume is performed

$$\frac{\partial}{\partial t}(\phi\rho_f) + \nabla\cdot(\phi\rho_f\boldsymbol{v}_f) = \rho_f q, \tag{2.17}$$

in which q is a volumetric flow rate, often associate with production or injection wells. If the compressibility of the solid and fluid are considered to be constants (MCTIGUE, 1986), Eq. 2.17 reduces to

$$\frac{\partial\theta}{\partial t} + \nabla\cdot\boldsymbol{v} = q, \tag{2.18}$$

Thus, by substituting Eq. 2.15 and Eq. 2.16 in Eq. 2.18, the following expression is achieved

$$\frac{1}{Q}\frac{\partial p}{\partial t} - \nabla\cdot\left(\frac{\boldsymbol{k}}{\mu}\nabla p\right) = q - \alpha\frac{\partial\varepsilon_v}{\partial t} - \nabla\cdot\left(\rho_f\frac{\boldsymbol{k}}{\mu}\boldsymbol{g}\right). \tag{2.19}$$

2.3 Micromechanical approach

Both geomechanics and single-phase flow models presented in this chapter relate the soil properties to its response, neither its components physical properties nor their individual responses were accounted for. Although this approach seems convenient, it constrains the bulk properties to a particular soil, without establishing any relation whatsoever between the physical constants of the bulk and its constituents.

In order to express the bulk properties in terms of its components, Detournay and Cheng performed a so-called "micromechanical analysis" of a poroelastic medium (DETOURNAY; CHENG, 1993). Assuming full saturation of the soil, the following expressions for α and $1/Q$ are achieved

$$\alpha = 1 - \frac{c_s}{c_b}, \tag{2.20}$$

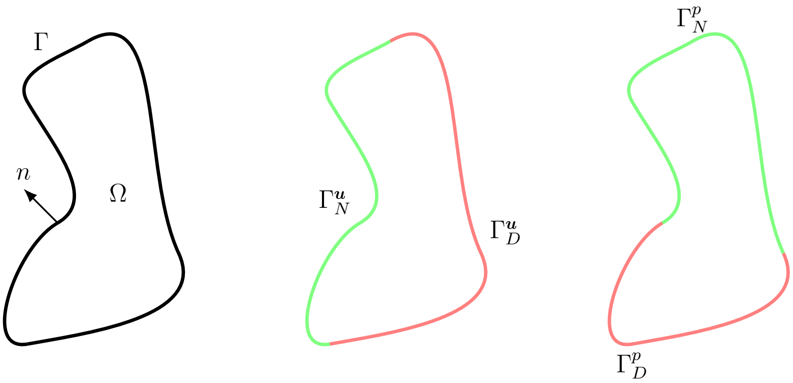
$$Q = \frac{1}{\phi c_f + (\alpha - \phi) c_s}, \tag{2.21}$$

where c_b, c_s and c_f stand for compressibility of the bulk, solid and fluid, in that order.

2.4 Boundary conditions

In order to close the mathematical model, the initial and boundary conditions must be elicited. The boundary surface Γ, whose unitary normal vector is $\boldsymbol{n}$, encompasses the domain Ω, where lays the porous medium. Furthermore, the boundary may be divided according to the applied boundary condition type. Thus, $\Gamma_N^{\boldsymbol{u}}$ and Γ_N^p denote surfaces where Neumann boundary conditions is imposed, while $\Gamma_D^{\boldsymbol{u}}$ and Γ_D^p regard surfaces subjected to Dirichlet boundary conditions — the superscript indicates which unknown is concerned.

Figure 2.1 – Domain Ω, surface boundary Γ and its subdivisions $\Gamma_N^{\boldsymbol{u}}$, $\Gamma_D^{\boldsymbol{u}}$, Γ_N^p, Γ_D^p.



Source: Own authorship.

At last, considering $\boldsymbol{r} = [x,\ y,\ z]^T$ the position vector, the unknowns to be determined are $\boldsymbol{u}(\boldsymbol{r}, t)$ and $p(\boldsymbol{r}, t)$ such that

$$\nabla_s^T(\boldsymbol{\sigma}' - \alpha p\boldsymbol{I}) + \rho\boldsymbol{g} + \nabla_s^T(\boldsymbol{\sigma}_0 + \alpha p_0\boldsymbol{I}) = 0 \quad \forall\boldsymbol{r} \in \Omega \tag{2.22}$$

$$\frac{1}{Q}\frac{\partial p}{\partial t} + \alpha\frac{\partial \varepsilon_v}{\partial t} - \nabla\cdot\left[\frac{\boldsymbol{k}}{\mu}\cdot(\nabla p - \rho_f\boldsymbol{g})\right] - q = 0 \quad \forall\boldsymbol{r} \in \Omega \tag{2.23}$$

$$(\boldsymbol{\sigma}' - \alpha p\boldsymbol{I})\cdot\hat{\boldsymbol{n}} = \boldsymbol{t} \quad \forall\boldsymbol{r} \in \Gamma_N^{\boldsymbol{u}} \tag{2.24}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \forall\boldsymbol{r} \in \Gamma_D^{\boldsymbol{u}} \tag{2.25}$$

$$-\boldsymbol{v}\cdot\hat{\boldsymbol{n}} = \dot{\omega} \quad \forall\boldsymbol{r} \in \Gamma_N^{p} \tag{2.26}$$

$$p = \bar{p} \quad \forall\boldsymbol{r} \in \Gamma_D^{p} \tag{2.27}$$

$$p(\boldsymbol{r}, 0) = p_0 \quad \forall\boldsymbol{r} \in \Omega \tag{2.28}$$

$$\boldsymbol{u}(\boldsymbol{r}, 0) = \boldsymbol{u}_0 \quad \forall\boldsymbol{r} \in \Omega, \tag{2.29}$$

in which $\boldsymbol{t}$ is the surface traction vector, $\dot{\omega}$ is the volumetric flux per unit area. The prescribed displacement and pressure are, in that order, $\bar{\boldsymbol{u}}$ and $\bar{p}$. Also, $\boldsymbol{u}_0$ and p_0 denote the displacement and pressure, respectively, at the initial state $t = 0$.

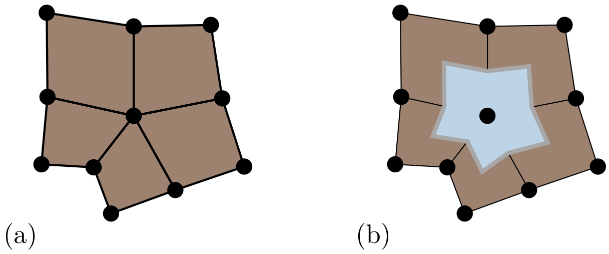
3 Computational grid

The spatial and time discretizations of the domain supports the approximations performed by a numerical method. Thus, to understand how this discretization is made contributes to the overall comprehension of the method. This chapter aims to show the fundamental geometrical entities of an unstructured grid, besides demonstrating how they are numerically exploited.

3.1 Geometrical aspects

As the EbFVM is a cell-vertex method, the unknowns are computed in the grid vertices. Therefore, the control volumes must be built around these entities, which is pictured in Fig. 3.1. This procedure is thoroughly explained by (HURTADO, 2012), however, a few remarks shall be made in the present work.

Figure 3.1 – Elements (a) and control volume (b).



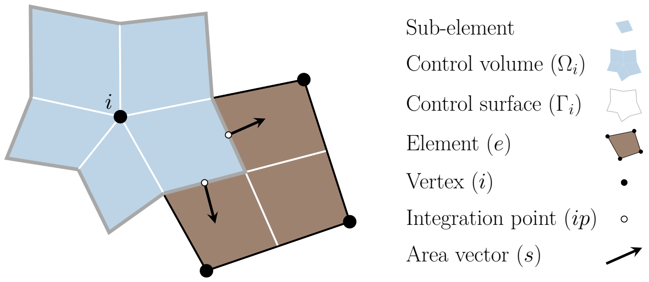
Source: Own authorship.

In order to obtain the control volumes, it is necessary to divide the grid elements into sub-elements, which are associated to each element vertex. The union of all sub-elements that share a common vertex i composes a control volume Ω_i. Furthermore, these entities are bounded by a control surface Γ_i, whose faces are characterized by an integration point ip, on its midpoint, and an area vector s, that points outwards Ω_i. For a two-dimensional grid, all entities mentioned in this paragraph may be visualized in Fig. 3.2.

For a three-dimensional unstructured grid, the same geometrical entities displayed in Fig. 3.2 are present. However, as the spatial visualization of these is trickier, the control volume building procedure shall be gradually performed. Hence, the concept of outer and inner faces is introduced, followed by the definition of sub-element.

The element vertices are arranged in collections to form facets, as portrayed in Fig. 3.3. These entities are also divided, so to build the outer faces, which in turn are related to each vertex that comprises the facet. Even though every sub-element is enveloped

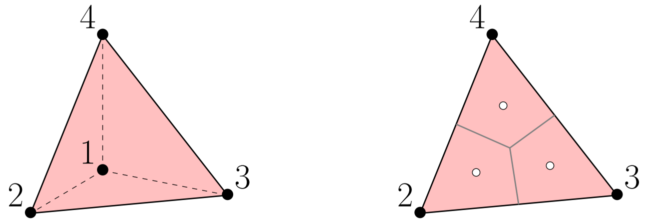
Figure 3.2 – Geometrical entities of a two-dimensional grid.



Source: Own authorship.

by outer faces, the latter only belong to a control surface if they are located on the domain boundaries. Therefore, outer faces are only significant to boundary control volumes, being disregarded otherwise.

Figure 3.3 – Tetrahedron and its facets.



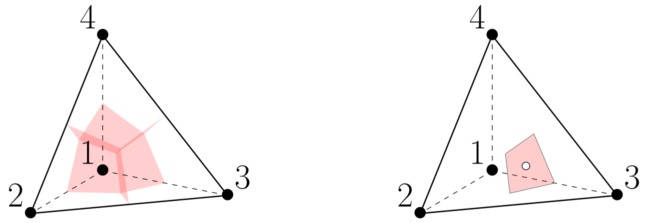
Facet	Vertices
1	1, 3, 2
2	1, 2, 4
3	2, 3, 4
4	3, 1, 4

(a) Tetrahedron. (b) Facet 3 and its outerfaces. (c) Facet definition.

Source: Own authorship.

Furthermore, elements also have inner faces, which are associated to pairs of adjacent vertices, as shown in Fig. 3.4. Unlike outer faces, inner faces always belong to a control surface. Since each element is the unique owners of its inner faces, all computations regarding the latter are performed through a loop across the domain elements. This is quite convenient, considering that no searching algorithms are required in order to compute fluxes between control volumes, as in most cell centre methods.

Figure 3.4 – Tetrahedron inner faces.



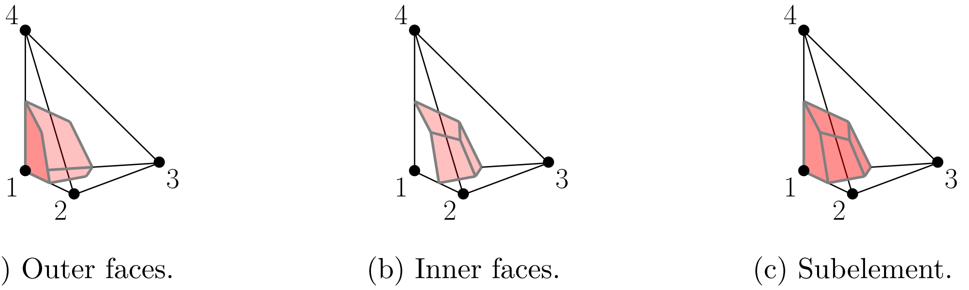
Inner face	Vertices
1	1, 2
2	2, 3
3	3, 1
4	1, 4
5	2, 4
6	3, 4

(a) Inner faces. (b) Inner face 3. (c) Inner face definition.

Source: Own authorship.

In turn, the sub-element related to an element vertex is defined as the volume enveloped by the surface composed by the union of all faces that share this vertex, as displayed in Fig. 3.5.

Figure 3.5 – Subelement composition.



(a) Outer faces.

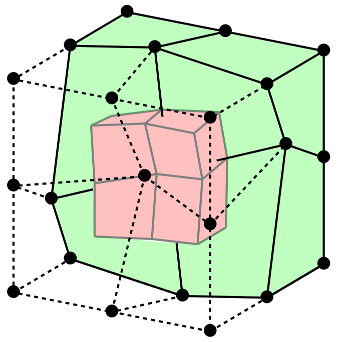
(b) Inner faces.

(c) Subelement.

Source: Own authorship.

Finally, the control volume Ω_i is characterized as the union of all sub-elements $\dot{s}$ associated to vertex i. Hereafter, this set of sub-elements shall be denoted as $\mathscr{H}_i$. In addition, Fig. 3.6 illustrates a control volume regarding an unstructured hexahedra grid.

Figure 3.6 – Control volume.



Source: Own authorship.

As remarked throughout this section, there is a unique correspondence between several geometrical entities and the element vertices. For example, the inner face 1 of a tetrahedron is connected to its vertices 1 and 2. Therefore, the ordering of these entities within an element must be consistent, so to ensure the subsequent associations are accurate.

For this reason, it was chosen to follow the convention established in the open-source library *CFD General Notation System* (POIRIER et al., 1998), as it is widely employed by numerical simulation softwares. Furthermore, the CGNS provides several routines regarding data input/output, also supporting parallel I/O (HAUSER, 2004).

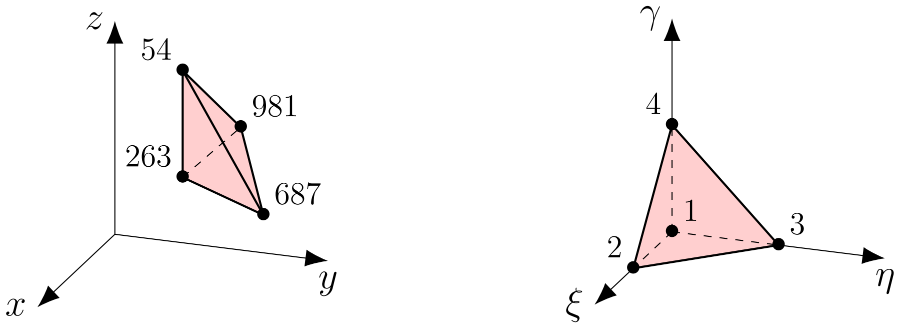
3.2 Numerical aspects

The geometrical entities described in the previous section are exploited in the numerical approximations carried out by the EbFVM. Besides, computing areas and volumes, it is necessary to interpolate variables and approximate gradients all over the computational domain. Thus, to elicit how these approximations are done, the coordinate transformation operation shall be first introduced, followed by the scalar field interpolation and gradient approximation procedures.

3.2.1 Coordinate transformation

The geometrical flexibility provided by unstructured grids comes at the price of irregularity, *i.e*, the elements often are irregular, and its dimensions are not known beforehand. To bypass this drawback, the global coordinates (x, y, z) of an element are transformed into local coordinates (ξ, η, γ) by a linear transformation. Therefore, the element becomes regular with fixed dimensions known in advance, as shown in Fig. 3.7.

Figure 3.7 – Coordinate transformation.



Source: Own authorship.

Hence, all computations within an element are performed in the transformed domain and then are transferred to the original domain through the element shape functions. These linear transformations relate the transformed domain coordinates of a point $\bar{\boldsymbol{r}} = (\xi,\ \eta,\ \gamma)$ to its original coordinates employing

$$x(\bar{\boldsymbol{r}}) = \sum_{i=1}^{m} N_i(\bar{\boldsymbol{r}}) x_i, \tag{3.1}$$

in which the number of element vertices is denoted by m, x_i represents the x coordinate of its vertex i, and N_i indicates the shape function associated with the element vertex i evaluated at $\bar{\boldsymbol{r}}$. Using the matricial notation, Eq. 3.1 may be rewritten as

$$x(\bar{\boldsymbol{r}}) = \boldsymbol{N}(\bar{\boldsymbol{r}}) \boldsymbol{x}_e, \tag{3.2}$$

where

$$\boldsymbol{N} = \begin{bmatrix} N_1 & \cdots & N_m \end{bmatrix}, \tag{3.3}$$

$$\boldsymbol{x}_e = \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix}^T. \tag{3.4}$$

When the position $\bar{\boldsymbol{r}}$ in the transformed domain is not specifically relevant, the term $\bar{\boldsymbol{r}}$ may be omitted. Consequently, Eq. 3.2 reduces to

$$x = \boldsymbol{N}\boldsymbol{x}_e. \tag{3.5}$$

3.2.2 Scalar field interpolation

The interpolation of a scalar field $\boldsymbol{\Phi}_e$, stored in the element vertices, follows the same procedure described in the previous subsection. Hence, the interpolation of $\boldsymbol{\Phi}_e$ at any point inside the element is given by

$$\Phi = \boldsymbol{N}\boldsymbol{\Phi}_e. \tag{3.6}$$

3.2.3 Gradient approximation

In order to evaluate the gradient of a function Φ within an element, it is necessary to approximate its derivatives at the desired position. Thus, the shape function derivatives are employed in this approximation, as in

$$\begin{bmatrix} \Phi_\xi \\ \Phi_\eta \\ \Phi_\gamma \end{bmatrix} = \begin{bmatrix} N_{1,\xi} & \cdots & N_{m,\xi} \\ N_{1,\eta} & \cdots & N_{m,\eta} \\ N_{1,\gamma} & \cdots & N_{m,\gamma} \end{bmatrix} \begin{bmatrix} \Phi_1 \\ \vdots \\ \Phi_m \end{bmatrix} = \boldsymbol{D}\boldsymbol{\Phi}_e, \tag{3.7}$$

in which $\boldsymbol{D}$ contains the shape function derivatives regarding the local coordinates.

Then, the inverse Jacobian matrix is used to transform the local derivatives of Φ into global derivatives,

$$\nabla\Phi = \begin{bmatrix} \Phi_x \\ \Phi_y \\ \Phi_z \end{bmatrix} = \begin{bmatrix} \xi_x & \eta_x & \gamma_x \\ \xi_y & \eta_y & \gamma_y \\ \xi_z & \eta_z & \gamma_z \end{bmatrix} \begin{bmatrix} \Phi_\xi \\ \Phi_\eta \\ \Phi_\gamma \end{bmatrix} = \boldsymbol{J}^{-T}\boldsymbol{D}\boldsymbol{\Phi}_e, \tag{3.8}$$

where $\boldsymbol{J}^{-T}$ is the transposed inverse of the coordinate transformation Jacobian matrix. The latter is given by

$$\boldsymbol{J} = \begin{bmatrix} x_\xi & x_\eta & x_\gamma \\ y_\xi & y_\eta & y_\gamma \\ z_\xi & z_\eta & z_\gamma \end{bmatrix} \quad \therefore \quad \boldsymbol{J}^T = \begin{bmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\gamma & y_\gamma & z_\gamma \end{bmatrix}, \tag{3.9}$$

being all derivatives numerically evaluated.

3.2.4 Gradient operator

As gradients are often computed throughout the computational domain, it is convenient to define the gradient operator $\boldsymbol{B}$, such that

$$\boldsymbol{B} = \boldsymbol{J}^{-T}\boldsymbol{D} = \begin{bmatrix} \partial_x \\ \partial_y \\ \partial_z \end{bmatrix} \begin{bmatrix} N_1 & \cdots & N_m \end{bmatrix} = \begin{bmatrix} N_{1,x} & \cdots & N_{m,x} \\ N_{1,y} & \cdots & N_{m,y} \\ N_{1,z} & \cdots & N_{m,z} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}_x \\ \boldsymbol{N}_y \\ \boldsymbol{N}_z \end{bmatrix}. \tag{3.10}$$

In this manner, the gradient of Φ may be recovered at an integration point through

$$\nabla\Phi_{pi} = \boldsymbol{B}_{pi}\boldsymbol{\Phi}_e. \tag{3.11}$$

Moreover, it is also useful to define the symmetric gradient operator $\boldsymbol{B}_s$, which is employed in the evaluation of the symmetric nabla operator ∇_s. However, it is first necessary to rearrange the element shape function values as

$$\boldsymbol{N}_s = \begin{bmatrix} \boldsymbol{N} & 0 & 0 \\ 0 & \boldsymbol{N} & 0 \\ 0 & 0 & \boldsymbol{N} \end{bmatrix}. \tag{3.12}$$

Thus, $\boldsymbol{B}_s$ is achieved by applying ∇_s to $\boldsymbol{N}_s$

$$\boldsymbol{B}_s = \nabla_s\boldsymbol{N}_s = \begin{bmatrix} \partial_x & 0 & 0 \\ 0 & \partial_y & 0 \\ 0 & 0 & \partial_z \\ \partial_y & \partial_x & 0 \\ 0 & \partial_z & \partial_y \\ \partial_z & 0 & \partial_x \end{bmatrix} \begin{bmatrix} \boldsymbol{N} & 0 & 0 \\ 0 & \boldsymbol{N} & 0 \\ 0 & 0 & \boldsymbol{N} \end{bmatrix} = \begin{bmatrix} \boldsymbol{N}_x & 0 & 0 \\ 0 & \boldsymbol{N}_y & 0 \\ 0 & 0 & \boldsymbol{N}_z \\ \boldsymbol{N}_y & \boldsymbol{N}_x & 0 \\ 0 & \boldsymbol{N}_z & \boldsymbol{N}_y \\ \boldsymbol{N}_z & 0 & \boldsymbol{N}_x \end{bmatrix}. \tag{3.13}$$

4 Numerical model

In this chapter, the discretization of the mathematical model equations is achieved through the EbFVM. Hence, the geometrical entities and the numerical tools introduced in Chap. 3 are finally exploited. Following the discretization of the geomechanical model equation 2.13, the algebraic form of the single-phase flow equation 2.19 is obtained.

4.1 Discretization procedure

The partial differential equations of the mathematical model are integrated over a control volume Ω_i, in order to obtain their discrete form. Thereupon, Gauss's divergence theorem is employed, so the volumetric integrals of the divergence terms become surface integrals over Γ_i — which requires the evaluation of mass and momentum fluxes across the control surface. In turn, these surface integrals are approximated through the midpoint rule, being thus reduced to the sum of fluxes that cross each integration point of Γ_i.

In addition, the computation of fluxes across an integration point requires the displacement and pressure, as well as their gradients, at the same position. These are approximated by

$$\begin{aligned} p_{pi} &\approx \boldsymbol{N}_{pi}\boldsymbol{p}_e & (4.1)\\ (\nabla p)_{pi} &\approx \boldsymbol{B}_{pi}\boldsymbol{p}_e & (4.2)\\ \boldsymbol{u}_{pi} &\approx (\boldsymbol{N}_s)_{pi}\boldsymbol{u}_e & (4.3)\\ (\nabla_s \boldsymbol{u})_{pi} &\approx (\boldsymbol{B}_s)_{pi}\boldsymbol{u}_e, & (4.4) \end{aligned}$$

where $\boldsymbol{N}, \boldsymbol{N}_s, \boldsymbol{B}, \boldsymbol{B}_s$ are the operators defined in Chap. 3. The vectors $\boldsymbol{p}_e$ and $\boldsymbol{u}_e$ store, in that order, the pressure and displacement belonging to the m vertices of an element e, such that

$$\boldsymbol{p}_e = \begin{bmatrix} p_1 & \cdots & p_m \end{bmatrix}^T, \tag{4.5}$$

$$\boldsymbol{u}_e = \begin{bmatrix} u_1 & \cdots & u_m & v_1 & \cdots & v_m & w_1 & \cdots & w_m \end{bmatrix}^T. \tag{4.6}$$

Furthermore, as in every control volume method, the fluxes that leave a control volume surface are equal to the fluxes entering its neighbour, so to ensure the conservation of all concerning physical quantities.

As a last remark concerning the EbFVM, the physical properties are defined within an element. Thus, when evaluating fluxes across an inner faces, no interpolations are necessary.

4.2 Geomechanical model

The geomechanical model equations represent a mechanical balance across the porous medium. Their integration over a control volume Ω_i leads to

$$\int_{\Omega_i} \nabla \cdot (\boldsymbol{\sigma}' - \alpha p \boldsymbol{I})\, d\Omega_i \;=\; -\int_{\Omega_i} \nabla \cdot (\boldsymbol{\sigma}_0 + \alpha p_0 \boldsymbol{I})\, d\Omega_i - \int_{\Omega_i} \rho \boldsymbol{g}\, d\Omega_i, \tag{4.7}$$

In order to better elicit the integration procedure, each term of Eq. 4.7 shall be individually treated. Thus, starting by the gravitational term, the gravity vector $\boldsymbol{g}$ is assumed to be constant over the control volume. On the other hand, the density ρ may vary across the sub-elements. Therefore, the body force integral reduces to

$$\int_{\Omega_i} \rho \boldsymbol{g}\, d\Omega_i \;=\; \boldsymbol{g} \int_{\Omega_i} \rho\, d\Omega_i \;=\; \boldsymbol{g} \sum_{\dot{s} \in \mathscr{H}_i} \rho_{\dot{s}} \Delta\Omega_{\dot{s}}, \tag{4.8}$$

in which $\Delta\Omega_{\dot{s}}$ and $\rho_{\dot{s}}$ are the sub-element volume and density, respectively.

In turn, the integrals regarding the divergence terms are evaluated through Gauss's theorem. Hence, the volumetric integrals become surface integrals over the control surface Γ_i, as in

$$\int_{\Omega_i} \nabla \cdot (\boldsymbol{\sigma}' - \alpha p \boldsymbol{I})\, d\Omega_i \;=\; \int_{\Gamma_i} (\boldsymbol{\sigma}' - \alpha p \boldsymbol{I}) \cdot \hat{\boldsymbol{n}}_s\, d\Gamma_i. \tag{4.9}$$

with $\hat{\boldsymbol{n}}_s$ being the symmetric normal unit vector.

Once that Γ_i may be divided into faces, the midpoint rule is employed in the evaluation of right-hand side of Eq. 4.9. Therefore, the latter becomes

$$\int_{\Gamma_i} (\boldsymbol{\sigma}' - \alpha p \boldsymbol{I}) \cdot \hat{\boldsymbol{n}}_s\, d\Gamma_i = \sum_{\text{ip} \in \Gamma_i} [(\boldsymbol{\sigma}' - \alpha p \boldsymbol{I}) \boldsymbol{s}_s]_{ip}, \tag{4.10}$$

where $\boldsymbol{s}_s$ is the symmetric face area vector. Considering the face area vector $\boldsymbol{s} = (s_x, s_y, s_z)^T$, $\boldsymbol{s}_s$ is given by

$$\boldsymbol{s}_s = \begin{bmatrix} s_x & 0 & 0 & s_y & 0 & s_z \\ 0 & s_y & 0 & s_x & s_z & 0 \\ 0 & 0 & s_z & 0 & s_y & s_x \end{bmatrix}^T . \tag{4.11}$$

At this stage, the integral regarding the initial state of the porous medium is fully evaluated, given the fact that $\boldsymbol{\sigma}_0$ and p_0 are known, thus

$$\int_{\Omega_i} \nabla \cdot (\boldsymbol{\sigma}_0 + \alpha p_0 \boldsymbol{I})\, d\Omega_i \approx \sum_{\mathrm{ip}\in\Gamma_i} [(\boldsymbol{\sigma}_0 + \alpha p_0 \boldsymbol{I})) \boldsymbol{s}_s]_{ip}. \tag{4.12}$$

However, further approximations must be done to express the remaining integral in terms of the nodal points. Hence, Eq. 4.1 and Eq. 4.4 are employed to evaluate, respectively, the pore pressure and the displacement vector at the integration points. Consequently, Eq. 4.10 becomes

$$\sum_{\mathrm{ip}\in\Gamma_i} (\boldsymbol{s}_s^T \mathbb{C}_e \boldsymbol{B}_s \boldsymbol{u}_e - \alpha \boldsymbol{s}_s^T \boldsymbol{N}_i \boldsymbol{p}_e)_{ip} = \sum_{\mathrm{ip}\in\Gamma_i} (\boldsymbol{s}_s^T \boldsymbol{\sigma}_0 - \alpha \boldsymbol{s}_s^T p_0)_{ip} - \boldsymbol{g} \sum_{\dot{s}\in\mathscr{H}_i} \rho_{\dot{s}} \Delta\Omega_{\dot{s}}. \tag{4.13}$$

Finally, the stiffness matrix $\boldsymbol{K}_{ip}$ and the pore pressure matrix $\boldsymbol{L}_{ip}$ may be defined at the integration points, so to keep a succinct notation. Thus, the final form of Eq. 4.13 is obtained

$$\sum_{\mathrm{ip}\in\Gamma_i} (\boldsymbol{K}_{ip} \boldsymbol{u}_e - \boldsymbol{L}_{pi} \boldsymbol{p}_e) = \sum_{\mathrm{ip}\in\Gamma_i} (\boldsymbol{s}_s^T [\boldsymbol{\sigma}_0 - \alpha p_0)]_{ip} - \boldsymbol{g} \sum_{\dot{s}\in\mathscr{H}_i} \rho_{\dot{s}} \Delta\Omega_{\dot{s}}. \tag{4.14}$$

4.3 Flow model

The flow model equation corresponds to the fluid phase mass balance. In order to obtain its discrete form, it has to be integrated over time and space. Beginning by the integration over a time interval between t and $t + \Delta t$,

$$\int_t^{t+\Delta t} \left(\frac{1}{Q}\frac{\partial p}{\partial t} + \alpha \frac{\partial \varepsilon_v}{\partial t} + \nabla \cdot \boldsymbol{v} - q \right) dt = 0. \tag{4.15}$$

Through the fundamental theorem of calculus, the terms regarding the pore pressure and the volumetric are effortlessly evaluated. Hence,

$$\int_t^{t+\Delta t} \left(\frac{1}{Q}\frac{\partial p}{\partial t} + \alpha \frac{\partial \varepsilon_v}{\partial t} \right) dt = \frac{1}{Q}(p - p^o) + \alpha(\varepsilon_v - \varepsilon_v^o), \tag{4.16}$$

where the superindex “o” indicates variables evaluated in the previous time step, and the variables computed in the current time step, $t + \Delta t$ carry no superindex whatsoever.

In turn, the remaining terms of Eq. 4.15 are estimated through a fully implicit backward first-order Euler scheme, as in

$$\int_t^{t+\Delta t} (\nabla \cdot \boldsymbol{v} - q)\, dt \approx \Delta t(\nabla \cdot \boldsymbol{v} - q). \tag{4.17}$$

Thus, dividing Eq. 4.16 and Eq. 4.17 by Δt and substituting them back in Eq. 4.15 leads to

$$\frac{1}{Q\Delta t}(p - p^o) + \frac{\alpha}{\Delta t}(\varepsilon_v - \varepsilon_v^o) + \nabla \cdot \boldsymbol{v} \; = \; q. \tag{4.18}$$

Once the time integrals have been evaluated, it is now necessary to integrate Eq. 4.18 over a control volume Ω_i. Recalling that $\varepsilon_v = \nabla \cdot \boldsymbol{u}$, this integration leads to

$$\int_{\Omega_i} \frac{p - p^o}{Q\Delta t} \, d\Omega_i + \int_{\Omega_i} \nabla \cdot \left[\boldsymbol{v} + \frac{\alpha}{\Delta t}(\boldsymbol{u} - \boldsymbol{u}^o) \right] d\Omega_i \; = \; \int_{\Omega_i} q \, d\Omega_i. \tag{4.19}$$

As done before, each term of Eq. 4.19 shall be separately treated. Starting by the source term, given that its value is constant inside Ω_i, it follows that

$$\int_{\Omega_i} q \, d\Omega_i = q_i \int_{\Omega_i} d\Omega_i = q_i \Delta\Omega_i. \tag{4.20}$$

where q_i is the volumetric flow rate at the control volume, which shall be evaluated in the next section.

The integration of the pore pressure term is also elementary. Considering that Biot's modulus $1/Q$ may only vary across the sub-elements, the integral may be approximated as

$$\int_{\Omega_i} \frac{p - p^o}{Q\Delta t} \, d\Omega_i \approx \frac{p_i - p_i^o}{\Delta t} \sum_{\dot{s} \in \mathscr{H}_i} \frac{\Delta\Omega_{\dot{s}}}{Q_{\dot{s}}}. \tag{4.21}$$

Furthermore, Gauss's theorem and the midpoint rule are employed in the evaluation of the divergence term of Eq. 4.19. Consequently, it reduces to

$$\int_{\Omega_i} \nabla \cdot \left[\boldsymbol{v} + \frac{\alpha(\boldsymbol{u} - \boldsymbol{u}^o)}{\Delta t} \right] d\Omega_i \approx \sum_{\mathrm{ip} \in \Gamma_i} \left[\boldsymbol{v}_{ip} + \frac{\alpha}{\Delta t}(\boldsymbol{u}_{ip} - \boldsymbol{u}_{ip}^o) \right] \boldsymbol{s}_{ip}, \tag{4.22}$$

where the displacements at the integration points are evaluated through Eq. 4.3, and the Darcy's velocity at the same position is approximated by

$$\boldsymbol{v}_{ip} = -\frac{\boldsymbol{k}_{ip}}{\mu}(\nabla p_{ip} - \rho_f \boldsymbol{g}) \approx -\frac{\boldsymbol{k}_e}{\mu}(\boldsymbol{B}_{ip}\boldsymbol{p}_e - \rho_f \boldsymbol{g}). \tag{4.23}$$

Hence, the substitution of Eq. 4.3 and Eq. 4.23 at the right-hand side of Eq. 4.22 results in

$$\sum_{\mathrm{ip} \in \Gamma_i} \left[-\frac{1}{\mu} \boldsymbol{s}_{ip}^T \boldsymbol{k}_e \boldsymbol{B}_{ip} \boldsymbol{p}_e + \frac{1}{\mu} \boldsymbol{s}_{ip}^T \boldsymbol{k}_e \rho_f \boldsymbol{g} + \frac{\alpha}{\Delta t} \boldsymbol{s}_{ip}^T (\boldsymbol{N}_s)_{ip} (\boldsymbol{u}_e - \boldsymbol{u}_e^o) \right]. \tag{4.24}$$

In order to keep a concise notation, it is useful to define the Darcy's velocity matrix $\boldsymbol{H}_{ip}$, the Darcy's gravity matrix $\boldsymbol{G}_{ip}$, and the volumetric strain matrix $\boldsymbol{Q}_{ip}$. Then, Eq. 4.24 may be rewritten as

$$\sum_{\mathrm{ip}\in\Gamma_i}\left[\boldsymbol{H}_{ip}\boldsymbol{p}_e + \boldsymbol{G}_{ip}\boldsymbol{g} + \boldsymbol{Q}_{ip}\boldsymbol{u}_e - \boldsymbol{Q}_{ip}\boldsymbol{u}_e^o)\right], \tag{4.25}$$

At last, the discrete form of the flow model equation is achieved by substituting Eq. 4.20, Eq. 4.21 and Eq. 4.25 at Eq. 4.19, which leads to

$$\frac{p_i}{\Delta t}\sum_{\dot{s}\in\mathscr{H}_i}\frac{\Delta\Omega_{\dot{s}}}{Q_{\dot{s}}} + \sum_{\mathrm{ip}\in\Gamma_i}(\boldsymbol{H}_{ip}\boldsymbol{p}_e + \boldsymbol{Q}_{ip}\boldsymbol{u}_e) = q_i\Delta\Omega_i + \frac{p_i^o}{\Delta t}\sum_{\dot{s}\in\mathscr{H}_i}\frac{\Delta\Omega_{\dot{s}}}{Q_{\dot{s}}} - \sum_{\mathrm{ip}\in\Gamma_i}(\boldsymbol{G}_{ip}\boldsymbol{g} - \boldsymbol{Q}_{ip}\boldsymbol{u}_e^o). \tag{4.26}$$

4.4 Well representation

The volumetric flow rate q_i is associated with control volumes that are intercepted by a well. In this scenario, the mass flux across the well walls may be written as

$$q_i = \frac{WI}{\mu}(p_i - p_w + h_i), \tag{4.27}$$

where p_i is the control volume pressure, p_w is the well top pressure, and WI is the well index concerning Ω_i. Furthermore, h_i is the difference in hydrostatic pressure between the well top and the control volume, that is,

$$h_i = (z_{top} - z_i)\rho_f\boldsymbol{g}, \tag{4.28}$$

where z_{top} and z_i are the vertical positions of the well top and Ω_i, respectively.

Regarding WI, its evaluation is performed according to Peaceman's well model (PEACEMAN et al., 1978),

$$WI = \frac{2\pi k\Delta z_i}{\ln\left(r_{eq}/r_w\right)}, \tag{4.29}$$

in which k is the absolute permeability, and r_w is the wellbore radius. Considering the control volume to be cubic, and that its dimensions are Δx, Δx and Δz, its equivalent radius r_{eq} is given by

$$r_{eq} = 0.28\sqrt{\Delta x^2 + \Delta y^2}. \tag{4.30}$$

Moreover, Eq. 4.27 introduces one more unknown, the well pressure p_w, which requires an additional equation to elicit its value. Thus, a mass balance on the well leads to

$$\sum_{i \in \Lambda} q_i = q_t, \tag{4.31}$$

with Λ being the set of control volumes intercepted by the well, and q_t is the total volumetric flow across its walls, whose value is prescribed. Therefore, the extra equation employed to evaluate p_w is given by

$$\sum_{i \in \Lambda} \frac{WI}{\mu}(p_i - p_w + h_i) = q_t. \tag{4.32}$$

5 Coupling schemes

The mathematical coupling between the geomechanical and fluid flow models has been discussed in Chap. 2, where both formulations had been exhibited. In turn, this chapter has the purpose of elicitating the numerical treatment of the coupled problem. Hence, after introducing the linear systems regarding the discrete form of the model equations, two coupling techniques shall be presented: the Fully Implicit Method and the Fixed-Stress Split.

5.1 Linear systems

In Chap. 4, the discretization in time and space of the mathematical model equations, Eq. 2.13 and Eq. 2.19, has been performed. As a result, two discrete equations have been obtained, Eq. 4.14 and Eq. 4.26, respectively. The former regards the algebraic form of the geomechanical model and, when assembled for every control volume of the grid, leads to the following linear of system

$$\mathbb{K}\boldsymbol{U} + \mathbb{L}\boldsymbol{P} = \boldsymbol{b}^{\boldsymbol{u}}, \tag{5.1}$$

in which $\mathbb{K}$ and $\mathbb{L}$ are the global stiffness and pore pressure matrices, in sequence. Considering m to be the number of grid vertices, the dimensions of $\mathbb{K}$ and $\mathbb{L}$ are, in that order, $(3m \times 3m)$ and $(3m \times m)$. The vector $\boldsymbol{U}$ represents the displacements u, v, w of all grid vertices, while $\boldsymbol{P}$ contains the pore pressure regarding the same entities — their sizes are $(3m \times 1)$ and $(m \times 1)$, respectively. Also, the independent term $\boldsymbol{b}^{\boldsymbol{u}}$ accounts for the initial state of the domain, the body forces due the gravitational field, and possible boundary conditions.

Furthermore, Eq. 4.26 represents the discrete single-phase flow model and, if built for every grid vertex, the subsequent linear is obtained

$$\mathbb{Q}\boldsymbol{U} + (\mathbb{A} + \mathbb{H})\boldsymbol{P} = \boldsymbol{b}^{p}, \tag{5.2}$$

with $\mathbb{H}$ and $\mathbb{A}$ being, in that order, the global Darcy's velocity and accumulation matrices, both sized $(m \times m)$. The global volumetric strain matrix, whose dimensions are $(m \times 3m)$, is denoted by $\mathbb{Q}$. Additionally, the independent vector $\boldsymbol{b}^{p}$, besides containing the eventual boundary conditions and source terms, holds the Darcy's gravity term $\mathbb{G}^{p}$, as well as the accumulation and volumetric strain terms. It may be written as

$$\boldsymbol{b}^p = \mathbb{A}\boldsymbol{P}^o + \mathbb{Q}\boldsymbol{U}^o - \mathbb{G}^p, \tag{5.3}$$

The coupling between the two linear systems is quite obvious, given the fact that Eq. 5.1 requires the pore pressure distribution to determine the displacement fields, while Eq. 5.2 relies on the latter to evaluate the former. Since both equations cannot be separately solved, coupling schemes must be exploited.

5.2 Fully implicit method

The Fully Implicit Method consists of simultaneously solving Eq. 5.1 and Eq. 5.2 in a single linear system, as in

$$\begin{bmatrix} \mathbb{K} & \mathbb{L} \\ \mathbb{Q} & (\mathbb{A} + \mathbb{H}) \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{b}^u \\ \mathbf{b}^p \end{bmatrix}. \tag{5.4}$$

Thus, the coupling is implicitly treated in the linear system. Consequently, to obtain both displacement and pressure fields regarding a time step, Eq. 5.4 must be solved, within the tolerance of the linear system solver, only once. Since no iterative procedure is required, the FIM is equivalent to a direct solving method in respect to coupling a set of equations.

5.3 Fixed-Stress Split

In the Fixed-Stress Split, Eq. 5.1 and Eq. 5.2 are segregately solved and iteratively coupled. According to (KIM; TCHELEPI; JUANES, 2011), the mass conservation equations are solved first. Afterwards, the resulting pressure distribution is employed in the solution of the mechanical balance equations. This whole procedure is repeated until convergence is achieved, for each time step.

In order to individually solve Eq. 5.2, whose differential form may be written as

$$\frac{1}{Q}\frac{\partial p}{\partial t}^{j+1} + \nabla \cdot \boldsymbol{v}^{j+1} = q - \alpha \frac{\partial \epsilon_v}{\partial t}^{j+1}, \tag{5.5}$$

in which j is the iterative level, the volumetric strain terms must be explicitly evaluated. The FSS consists of keeping the total volumetric stress, $\sigma_v = \sigma_v' - \alpha p$, fixed between iterations. Given that the effective volumetric stress is defined by

$$\sigma_v' = K\epsilon_v, \tag{5.6}$$

in which K is the bulk modulus, it follows that

$$\sigma_v'^{j+1} - \alpha p^{j+1} = \sigma_v'^{j} - \alpha p^{j} \;\therefore\; \epsilon_v^{j+1} = \epsilon_v^{j} + \frac{\alpha}{K}(p^{j+1} - p^{j}). \tag{5.7}$$

The substitution of Eq. 5.7 in Eq. 5.5 leads to the mass conservation for the FSS,

$$(\frac{1}{Q} + \frac{\alpha^2}{K})\frac{\partial p}{\partial t}^{j+1} + \nabla \cdot \boldsymbol{v}^{j+1} = q - \alpha \frac{\partial \epsilon_v}{\partial t}^{j} + \frac{\alpha^2}{K}\frac{\partial p}{\partial t}^{j}. \tag{5.8}$$

Furthermore, the discrete form of Eq. 5.8 is also achieved through EbFVM. If the resulting algebraic equations are assembled for every control volume of the grid, the following linear system is obtained

$$(\mathbb{A} + \mathbb{H} + \mathbb{D})\boldsymbol{P}^{j+1} = \boldsymbol{b}^{p} + \mathbb{D}\boldsymbol{P}^{o} - \mathbb{Q}\boldsymbol{U}^{j} + \mathbb{D}\boldsymbol{P}^{j}, \tag{5.9}$$

in which $\mathbb{D}$ is a diagonal matrix whose components are given by

$$\mathbb{D}_{ii} = \frac{\Delta\Omega_i}{\Delta t}\frac{\alpha^2}{K}. \tag{5.10}$$

At last, it is important to note that, if the convergence *criterion* employed by the iterative coupling method is rigorous enough, it must achieve the same solution as the FIM. This fact may be observed in Eq. 5.9, given that when $\boldsymbol{P}^{j+1} = \boldsymbol{P}^{j}$, it reduces back to Eq. 5.2.

6 Parallelization

This chapter has the purpose of eliciting the parallelization approach employed in the assemblage and solution of the coupled algebraic equations introduced in Chap. 5. Thus, the Domain Decomposition Method (DDM) is exhibited, followed by parallel procedures concerning the linear systems. Also, the open-source libraries that assisted the implementation of the parallel reservoir simulator are presented.

6.1 Domain Decomposition Method

In the Domain Decomposition Method, the computational grid is divided into smaller domains. Each of these is assigned to a processor, which becomes the only responsible to determine the former's unknown fields. By the time each processor is done evaluating the fields in its designated domain, all unknowns regarding the whole grid are determined. Given that all processors work concurrently, a drastic reduction in the total simulation time is expected.

Fundamentally, the domain is decomposed in two steps: the grid partitioning and the subdomain composition. Before introducing these procedures, it is useful to define the concepts of *partition* and *subdomain*. Beginning by the former,

Definition 6.1 (Partition)**.** A partition is a grouping of the domain vertices such that every vertex is included in only one partition.

Once that partitions don't share the vertices of the domain among each other, they are all *pairwise disjoint*, that is,

$$(\forall A, B \in \Omega)\ A \neq B \rightarrow A \cap B = \varnothing. \tag{6.1}$$

Furthermore, each partition originates a single subdomain. Subdomains are necessary given the operations regarding a numerical method — these approximations require not only vertices but several geometrical entities regarding the domain. Hence, it follows the definition

Definition 6.2 (Subdomain)**.** A subdomain is a set of geometrical entities that are associated to at least one vertex of a partition.

Therefore, all vertices that belong to a partition belong to its respective subdomain as well. Nevertheless, considering that most geometrical entities described in Chap. 3 are related to more than one vertex, a subdomain also has vertices that do not belong to its parent partition. Thus, two more definitions regarding a subdomain are of use

Definition 6.3 (Local Vertex)**.** If a vertex contained in a subdomain also belongs to the partition that originated it, the vertex is said to be local to the subdomain.

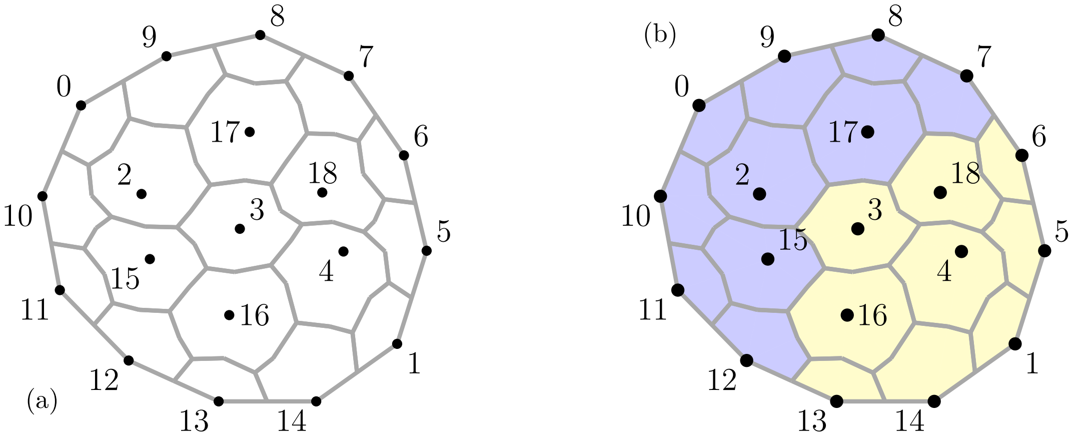
Definition 6.4 (Ghost Vertex)**.** If a vertex contained in a subdomain does NOT belong to the partition that originated it, the vertex is said to be ghost.

In consequence, a vertex can only be local to a single subdomain, since a partition is the unique owner of its vertices and it originates just one subdomain. Moreover, each subdomain is assigned to a processor, and it exclusively evaluates the unknowns concerning the local vertices of the subdomain. Thus, every control volume of the grid has its unknowns evaluated by a single processor.

6.1.1 Grid partitioning

For decomposing the domain into subdomains, the domain has to be partitioned, given that partitions are the foundation of the subdomains. Hence, the grid vertices are grouped to compose these partitions, which is portrayed in Fig. 6.1. As the control volumes belonging to each of the partitions shall have its unknowns evaluated by a single processor, this partitioning procedure determines how the computational effort is distributed among the processors.

Figure 6.1 – Domain (a) and its partitions (b).



Source: Own authorship.

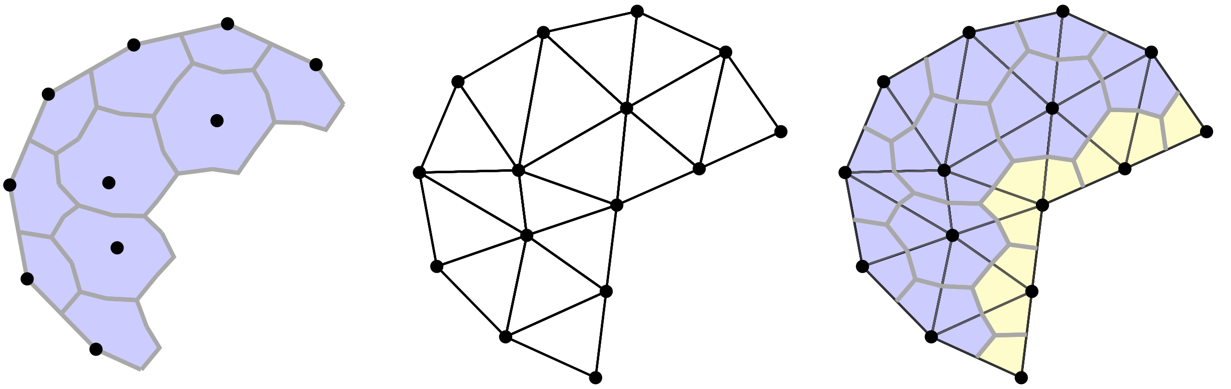
In order to maximise the parallel performance, the DDM must meet two requirements, both achieved in the grid partition stage. First, it has to assign nearly the same number of control volumes to each partition, as to better distribute the computational effort among the processors. In turn, the second requirement is to minimise the number of adjacent vertices attributed to different partitions, as this requires data to be exchanged among the processors — which is computationally costly. Therefore, the goal of the second condition is to reduce the overall communication.

Both of these requisites are met through *METIS* (KARYPIS; KUMAR, 2013), as it provides several algorithms for partitioning unstructured grids. To make use of these routines, a list containing the adjacency of each vertex of the grid must be supplied, together with the desired number of partitions. Finally, METIS outputs a list containing the partition which each vertex belongs to.

6.1.2 Subdomain composition

In what follows, the subdomain is formed by the union of all geometrical entities that have at least one vertex owned by the partition. Even though some of these entities have vertices that do not belong to the partition, the respective subdomain contains these control volumes, as illustrated in Fig. 6.2. These vertices are the so-called ghost vertices, recalling that no unknowns regarding them are evaluated in the subdomain.

Figure 6.2 – Subdomain composition.



(a) Partition.

(b) Elements that share at least one vertex owned by the partition.

(c) Subdomain. The sub-elements associated to ghost vertices are painted in yellow.

Source: Own authorship.

Given that some of these geometrical entities may have vertices belonging to several partitions, they can be shared among subdomains. Thus, the latter may have elements in common, which happens every time adjacent vertices are designated to distinct partitions — this is denominated *edge-cutting*. As already mentioned, it requires communication among processors and has to be minimised during the grid partitioning procedure.

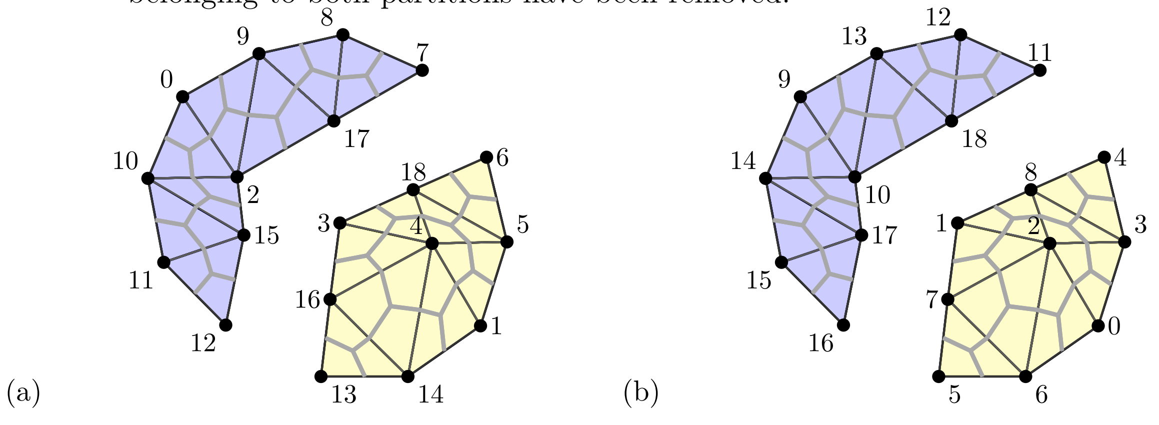
Another consequence of edge-cutting is the presence of ghost vertices, the only goals of these vertices are to promote data exchange among processors and offer geometrical support to local vertices. Regarding the data exchange, it must occur, for example, in the interpolation of a scalar field at an integration point. As this procedure requires the field values at the element vertices, the values concerning the ghost vertices are necessary. In turn, the geometrical support provided by the ghost vertices is needed, for instance, in the evaluation of the gradient operator $\boldsymbol{B}_{pi}$ — given that it requires the coordinates of all element vertices.

6.1.3 Vertex indexation

In the DDM, each processor determines unknowns only within its assigned subdomain local vertices. However, this evaluation requires data to be exchanged among the processors, given that what happens in a subdomain affects the others, and vice versa. As already discussed, this exchange of data is performed through the ghost vertices, which may belong to several subdomains. Thus, it would be practical to have a common vertex indexation throughout the subdomains. Therefore, this subsection aims to elicit how such numbering is obtained, besides discussing an intrinsic indexation to each subdomain.

After the grid partitioning stage, the domain vertices are renumbered so that their indices are contiguous within a partition. For example, the partitions 0 and 1, pictured in Fig. 6.1b, have 9 and 10 control volumes, respectively. Then, the vertices are reordered such that the first partition contains vertices 0 until 8, while the second partition holds vertices 9 until 18. This procedure is illustrated in Fig. 6.3, where Fig. 6.3a shows the original indexation, which came from the grid generator, and Fig. 6.3b shows the reordered indices, hereafter referred to as *global indices*. The purpose of this task is to facilitate the exchanges of data among the processors, as well as the linear system assemblage, given that the global indexation is shared by all subdomains.

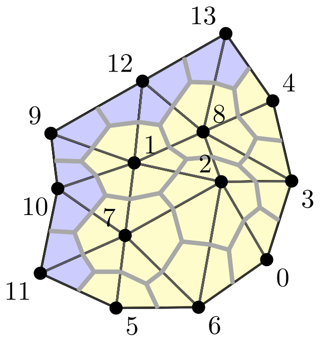
Figure 6.3 – Indexation: original (a) and global (b). The elements that have vertices belonging to both partitions have been removed.



Source: Own authorship.

When the domain portrayed in Fig. 6.1a is decomposed in two, the resulting subdomains may be visualized in Fig. 6.4 and Fig. 6.5. Both figures display the *local indices* of the control volumes, being this indexation internal to a subdomain. Considering, for instance, the yellow partition in Fig. 6.1b and its subdomain, shown in Fig. 6.4. The former has 9 vertices, while the latter has 14 — accounting for both local and ghost. Thus, the local indices of the subdomain control volumes shall range from 0 to 13, being from 0 to 8 local vertices, and the remaining being ghost vertices.

Figure 6.4 – Processor 0 subdomain.



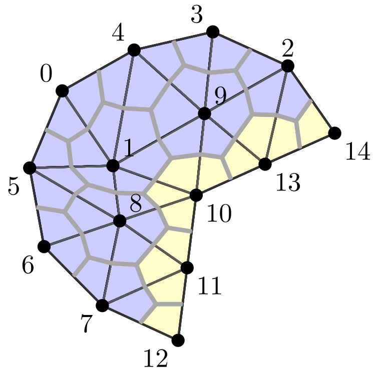
Local	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Global	0	1	2	3	4	5	6	7	8	10	17	16	18	11
Original	1	3	4	5	6	13	14	16	18	2	15	12	17	7

Source: Own authorship.

This indexation is useful for storing field values concerning a subdomain because it allows both local and ghost values to be kept in the same contiguous data structure. For example, considering the subdomain portrayed in Fig. 6.5, and that the pressure regarding all its vertices is stored in a C like array. The entries of the array ranging from 0 to 9 consists of local values, which must be exported as a simulation result, and the remaining entries hold ghost values, which must be received from another subdomain.

Therefore, this local indexation does not require local and ghost data to be kept in distinct structures. As shall be demonstrated in Sec. 6.2, this is quite convenient.

Figure 6.5 – Processor 1 subdomain.



Local	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Global	9	10	11	12	13	14	15	16	17	18	1	7	5	8	4
Original	0	2	7	8	9	10	11	12	15	17	3	16	12	18	6

Source: Own authorship.

6.1.4 DivideEtImpera

The three tasks – grid partitioning, subdomain composition and vertex indexation – related to the Domain Decomposition Method are performed by an open-source library called *DivideEtImpera* (GIACOMELLI, 2019), as it was developed to offer the DDM to numerical simulators that employ the cell-vertex scheme. Consequently, DEI was promptly integrated into EFVLib2018, a few modifications in its core allowed the parallel exploitation of its geomechanics-flow features.

Moreover, DivideEtImpera interfaces to the parallel I/O features of CGNS, in order to allow the simulation results to be written in parallel. Hence, each processor outputs only the unknowns concerning the local control volumes of its assigned subdomain.

6.2 Linear system assemblage

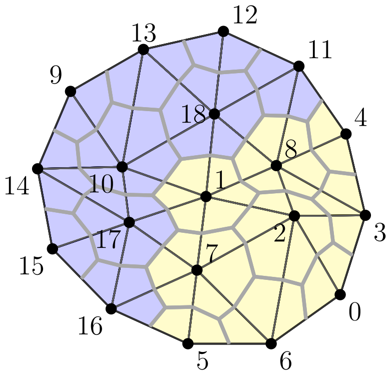
The linear system is also divided among the processors, being each of these responsible for a contiguous chunk of rows concerning the independent and solution vectors, as well as the coefficient matrix.

Hereafter, this block of rows shall be called *ownership range* and its determination is quite straightforward. Its size is equal to the number of local vertices owned by the subdomain times the amount of degrees of freedom per vertex to be computed. For example, considering the domain portrayed in Fig. 6.6, and that there are 3 degrees of freedom per vertex to be determined, processors 0 and 1 would have ownership ranges sized 27 and 30, respectively. In addition, the former would be responsible for the assemblage of rows 0 to 26, as the latter would assemble rows 27 to 56.

In the same manner that a processor only evaluates unknowns regarding the local vertices of its designated subdomain, it should only modify rows that are within its ownership range. Otherwise, additional communication between processors would be required, therefore degrading performance. The only exception to this rule is in the presence of a well, as this renders an additional equation to be solved, in accordance with Sec. 4.4. In this scenario, given that the well may be distributed throughout several subdomains, distinct processors would have to modify the same row.

For handling the parallel assemblage of the linear system, the *Portable, Extensible Toolkit for Scientific Computation* (BALAY et al., 2019) was employed. Thus, its data structures were used to store the coefficient matrix, as well as the independent and solution vectors. Besides being thoroughly tested and well documented, PETSc offers several routines for solving and preconditioning linear systems, which were also utilised in this work.

Figure 6.6 – Global indexation of the control volumes.



Source: Own authorship.

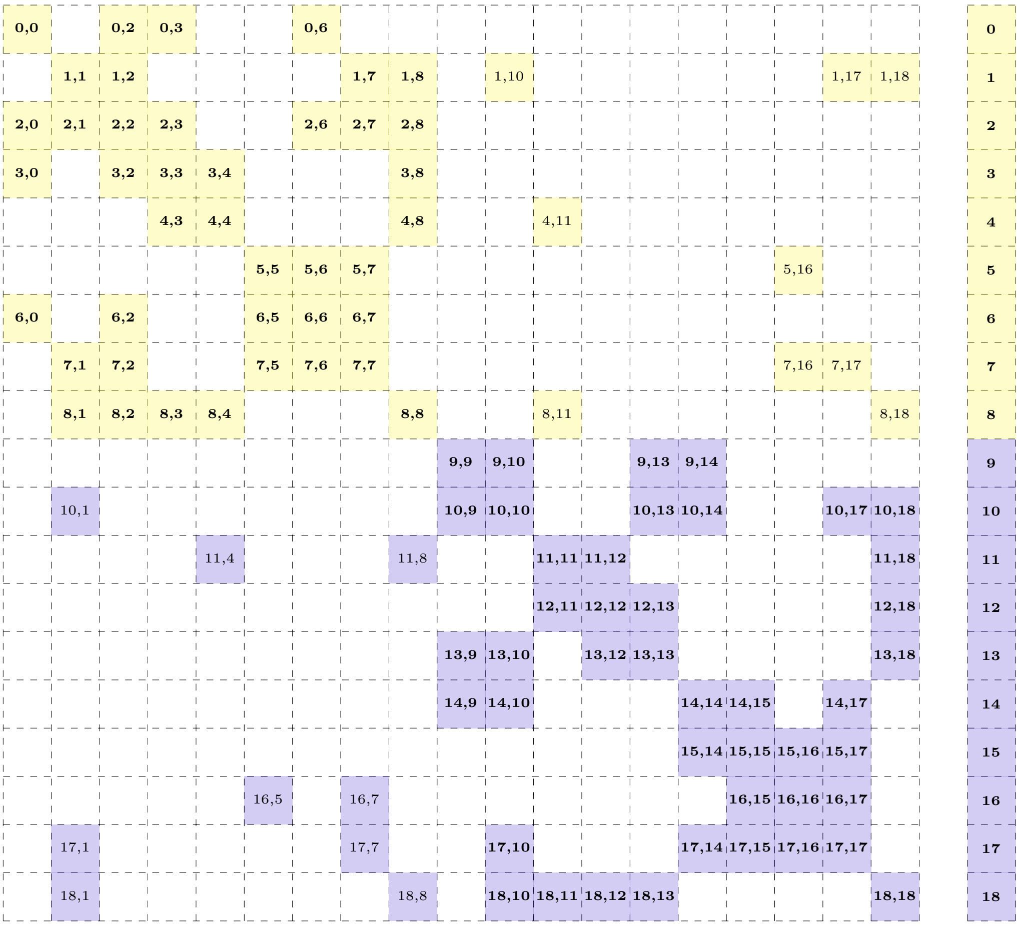
To illustrate this procedure, the grid shown in Fig. 6.6 is considered — the global indexation of the vertices is pictured. If the simulation of a single-phase flow in a porous medium is contemplated, there are only one degree of freedom to be solved. Hence, the assemblage of the operator $\boldsymbol{H}_{ip}$, the Darcy's velocity matrix, yields the linear system displayed in Fig. 6.7.

After the solution of the linear system, each processor only has the unknowns regarding its subdomain local vertices. Thus, the ghost values must be communicated to the former, as they may be eventually required. For instance, if processor 0 were to evaluate the pressure gradient within control volume 8, which is local to its subdomain, the pressure at vertices 11 and 18 would be needed.

In order to communicate ghost values among the subdomains, PETSc routines are employed once again. Given that PETSc data structures are utilised, this procedure becomes rather simple. As a mean of illustrating it, Fig. 6.8 shows the solution vectors obtained by processors 0 and 1 immediately after the linear system has been solved.

Then, the routine responsible for updating the ghost values is called, yielding the

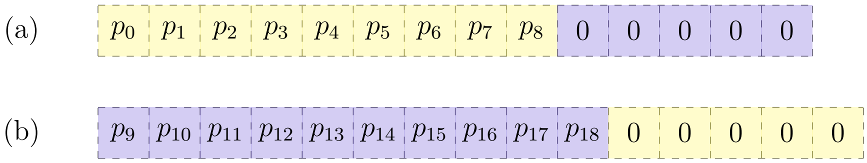
Figure 6.7 – Matrix.



0,0		0,2	0,3			0,6													0
	1,1	1,2					1,7	1,8		1,10							1,17	1,18	1
2,0	2,1	2,2	2,3			2,6	2,7	2,8											2
3,0		3,2	3,3	3,4				3,8											3
			4,3	4,4				4,8			4,11								4
					5,5	5,6	5,7									5,16			5
6,0		6,2			6,5	6,6	6,7												6
	7,1	7,2			7,5	7,6	7,7									7,16	7,17		7
	8,1	8,2	8,3	8,4				8,8			8,11							8,18	8
									9,9	9,10			9,13	9,14					9
	10,1								10,9	10,10			10,13	10,14			10,17	10,18	10
				11,4				11,8			11,11	11,12						11,18	11
											12,11	12,12	12,13					12,18	12
									13,9	13,10		13,12	13,13					13,18	13
									14,9	14,10				14,14	14,15		14,17		14
														15,14	15,15	15,16	15,17		15
					16,5		16,7								16,15	16,16	16,17		16
	17,1						17,7			17,10				17,14	17,15	17,16	17,17		17
	18,1							18,8		18,10	18,11	18,12	18,13					18,18	18

Source: Own authorship.

Figure 6.8 – Solution vectors of processors 0 (a) and 1 (b) before updating ghost values.

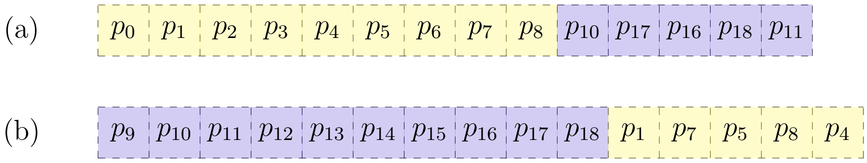


(a)	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	0	0	0	0	0
(b)	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	0	0	0	0	0

Source: Own authorship.

solution vectors displayed in Fig. 6.9.

Figure 6.9 – Solution vectors of processors 0 (a) and 1 (b) after updating ghost values.



(a)	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_{10}	p_{17}	p_{16}	p_{18}	p_{11}	
(b)	p_9	p_{10}	p_{11}	p_{12}	p_{13}	p_{14}	p_{15}	p_{16}	p_{17}	p_{18}	p_1	p_7	p_5	p_8	p_4

Source: Own authorship.

7 Results

The geometrical support necessary for EbFVM applications, described in Chap. 3, alongside the numerical operators of poroelasticity, presented in Chap. 4, were implemented with an in-house C++ library named EFVLib2018. After the integration of DivideEtImpera into the latter, all features of EFVLib2018 were parallelized, thus allowing the solution of several poroelasticity scenarios to be performed in parallel.

Following the validation of the algorithms implemented in EFVLib2018, a performance comparison between the Fully Implicit Method and the Fixed-Stress Split is carried out in two classical poroelasticity problems: the Terzaghi's column and the Cryer's sphere. Then, the most efficient coupling scheme is utilised in the simulation of a reservoir exploitation scenario.

7.1 Verification

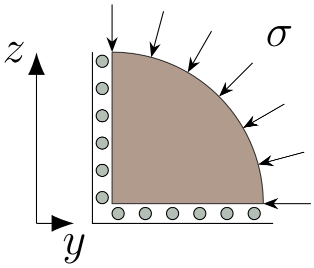
In order to verify the consistency of the numerical formulation implementation, the Cryer's sphere problem is solved, and the numerical values are compared against the analytical solution.

7.1.1 Cryer's sphere - analytical solution

The problem consists of a 1 m radius fully saturated poroelastic sphere. At $t = 0$, a load $\sigma = -1.00 \times 10^{+7}$ Pa is applied on its surface, where a zero pressure is prescribed, so to allow the fluid to freely leave the sphere. Considering the symmetries of this problem, the computational grid is reduced to only one eighth of the sphere, as illustrated in Fig. 7.1. On the flat surfaces pictured in the same figure, the pressure gradient is prescribed to 0, so there is no flow through them, and the normal displacement is prescribed to 0. Finally, the properties of the poroelastic medium and fluid are shown in Table 7.1 and Table 7.2, respectively.

For an unstructured tetrahedral grid composed of 5840 control volumes and a time step size of $\Delta t = 250$ s, the pressure at the centre of the sphere as displayed in Fig. 7.2. As it may be visualized, the numerical solution matches the analytical one, capturing the Mandel-Cryer effect (MANDEL, 1953) (HONÓRIO et al., 2019a). The latter is characterized by an increase in pressure, above the initially established, justified by the undrained conditions happening in the poroelastic medium centre. Furthermore, the Mandel-Cryer effect is an unmistakable evidence of the coupling between geomechanics and fluid flow, which uncoupled formulations fail to reproduce.

Figure 7.1 – Sketch of the Cryer's sphere.



Source: Own authorship.

Table 7.1 – Poroelastic properties of Charcoal Granite.

Property	Unit	Charcoal
ν	–	2.70×10^{-01}
G	Pa	$1.87 \times 10^{+10}$
c_s	Pa^{-1}	2.20×10^{-11}
ϕ	–	2.00×10^{-02}
k	m^2	1.00×10^{-19}

Source: (CHENG, 2016).

Table 7.2 – Fluid properties of water.

Property	Unit	Water
μ	$Pa \cdot s$	1.00×10^{-3}
c_f	Pa^{-1}	3.03×10^{-10}

Source: (CHENG, 2016).

Also, it is important to remark that the FSS achieved the same solution as the FIM when an absolute convergence *criterion* of 1 Pa was used.

Figure 7.2 – Center pressure over time.

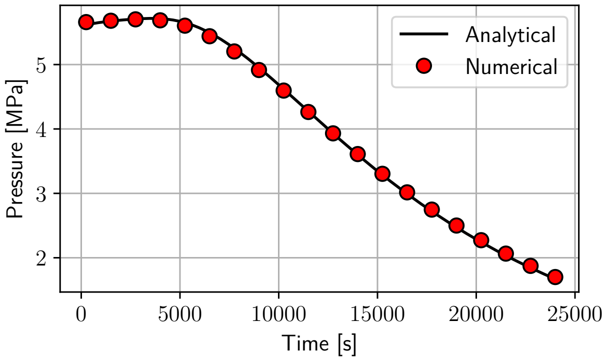
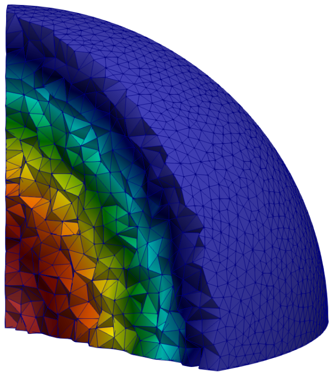


Figure 7.3 – Final pressure distribution.



7.2 Speedup comparison

The purpose of this work is to determine which coupling scheme yields the best performance for a parallel implementation of the DDM. Although the FIM outperforms the FSS in poroelasticity coupled with single-phase flow when a serial algorithm is exploited, little is known about the parallel performance of both. Besides determining which provides the largest speedup, an analysis regarding a dimensionless number and the amount of control volumes of the domain is carried out, so to ascertain if these parameters affect the parallel behaviour of the schemes. All results here shown were obtained through a Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz with 16,3 GB of RAM. Before displaying the results, a dimensionless parameter must be introduced.

7.2.1 Coupling intensity

The *coupling strength* (KIM; TCHELEPI; JUANES, 2011), hereafter referred to as *coupling intensity*, is denoted by τ and it measures how tight the geomechanics and fluid flow are coupled. As the coupling between these two *phenomena* increase, the iterative coupling schemes require more iterations to convergence. On the other hand, τ appears to show no influence on the FIM performance (HONÓRIO et al., 2019b). The coupling intensity is defined as

$$\tau = \frac{\alpha^2 Q}{K}, \tag{7.1}$$

where α is the Biot's coefficient, Q is the inverse of the Biot's modulus, and K is the bulk coefficient of soil.

As the coupling intensity changes, so does the time scale regarding the consolidation process. Thus, it is necessary to nondimensionalize the time, in order to keep the time scales consistent across the simulation. The dimensionless time (VERRUIJT, 2016) may be written as follows

$$t_d = \frac{c_v t}{l^2}, \tag{7.2}$$

in which t represents the time in real dimensions, l is a characteristic length, and c_v is the consolidation coefficient, given by

$$c_v = \frac{k}{\gamma_f(\alpha^2 m_v + 1/Q)}, \tag{7.3}$$

with γ_f being the fluid specific weight. The porous medium absolute permeability and confined compressibility are denoted by k and m_f, being the latter defined as follows

$$m_v = \frac{1}{K + \frac{4}{3}G}. \tag{7.4}$$

Unless otherwise stated, the final dimensionless time of the scenarios presented in this section is $t_d = 9.71 \times 10^{-8}$.

In addition, the numbers N_v and N_t shall denote the amount of control volumes of the domain and the number of time steps, in that order.

7.2.2 Linear system solution

For poroelasticity coupled with single-phase flow, the coefficient matrix A of the linear system has constant coefficients — that is the case for both FIM and FSS. Therefore, the LU-factorization could be applied to A only once, so to obtain the unknown vector x by just performing matrix products at each time step. This would be the fastest solving approach. Moreover, PETSc interfaces with the *SuperLU_DIST* library (LI; DEMMEL, 2003), thus easily allowing the usage of a parallel LU-factorization routine.

However, the coefficients of matrix A are only constant for the simplest case of coupled geomechanics models. If any nonlinearity were to be introduced, some coefficients of A would have to be recalculated in several occasions at each time step. Thus, solving the linear system through the LU-factorization would not be a good choice, performance wise.

Hence, the solution of the linear systems regarding the FIM was accomplished by the iterative solver *Generalized Minimal Residual* method (GMRES) available at PETSc. Given that the poroelasticity equations may yield very ill-conditioned linear systems, the use of preconditioners may be necessary. Therefore, the *Geometric Algebraic Multigrid* preconditioner (GAMG) implemented by PETSc is exploited alongside the GMRES.

In turn, for the FSS, the linear system concerning the geomechanics equations were solved by the LU-factorization, given that its coefficients are constant for linear elastic soils. On the other hand, the solution of the mass conservation linear system is achieved through the GMRES together with the *Successive Over Relaxation* preconditioner. In this case, the latter has shown to be more efficient than the GAMG.

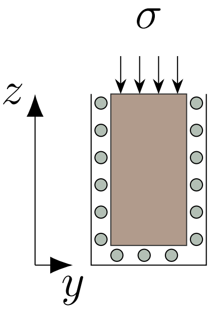
Finally, the matrices $\mathbb{A}$, $\mathbb{H}$, $\mathbb{D}$, that regard the mass conversation equations, are assembled at each time step. This is done in order to better weight the linear system assemblage in the overall simulation time.

7.2.3 Terzaghi's column

The next scenario involves a 6 m high poroelastic column with a 1 m^2 cross section area, as rendered in Fig. 7.4. Except for the top boundary, which has a constant load $\sigma = -1.00 \times 10^{+7}$ Pa applied and a prescribed pressure of 0, all boundaries of the column have zero normal displacement and are sealed.

In order to observe the speedup of FIM and FSS, a case in which $\tau = 7.51 \times 10^{-1}$ was initially performed. This coupling intensity was achieved through the poroelastic and fluid properties displayed in Table 7.3 and Table 7.6, in sequence. For a mixed element unstructured grid of 10770 vertices and a dimensionless time step of $\Delta t_d = 4.85 \times 10^{-7} = t_d/200$, the speedup of both coupling schemes are portrayed in Fig. 7.5. Even though the FSS achieved a larger speedup, 204% against 95% for 4 processors, it is still outperformed

Figure 7.4 – Sketch of the Terzaghi's column.



Source: Own authorship.

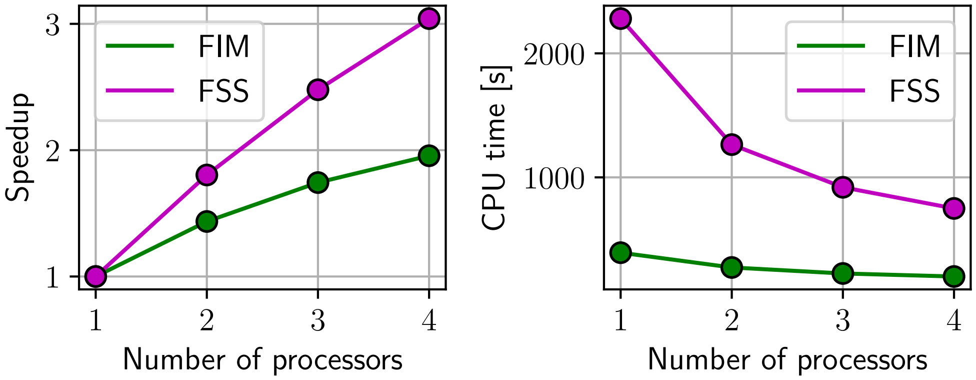
Table 7.3 – Poroelastic properties of Berea Sandstone.

Property	Unit	Berea
ν	$-$	2.00×10^{-01}
G	Pa	$6.00 \times 10^{+09}$
c_s	Pa^{-1}	2.78×10^{-11}
ϕ	$-$	1.90×10^{-01}
k	m^2	1.90×10^{-13}

Source: (CHENG, 2016).

by the FIM — the employed absolute convergence *criterion* was of 1.0×10^{-5} Pa.

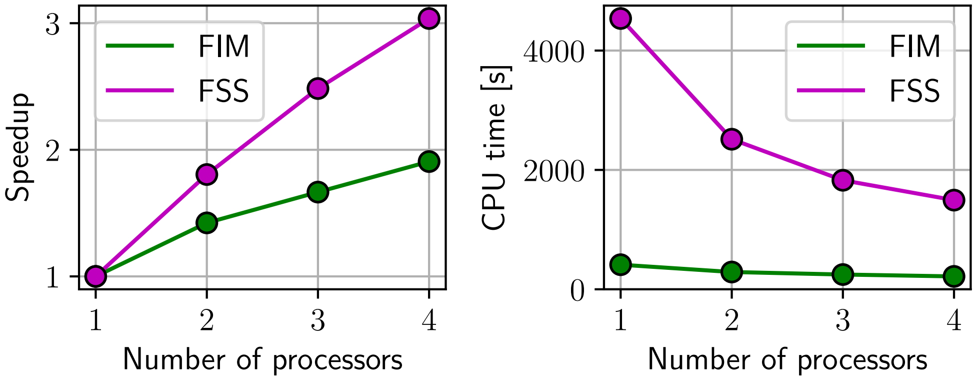
Figure 7.5 – Speedup and CPU time achieved in the Terzaghi's column problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 10770$, $N_t = 200$.



Source: Own authorship.

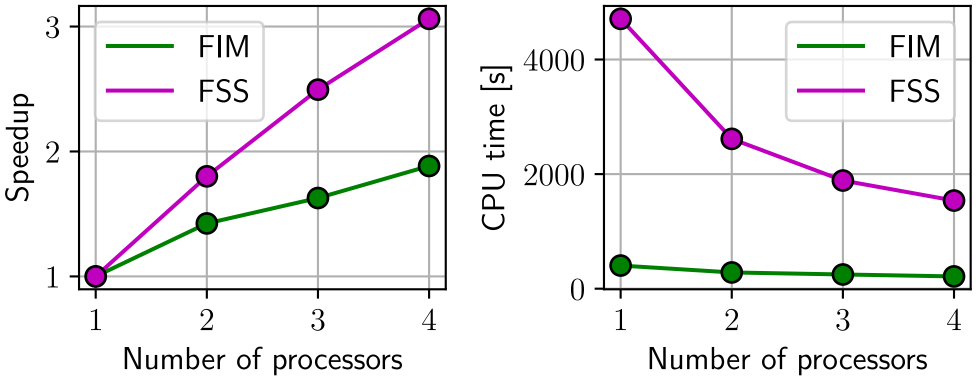
Increasing the coupling intensity to $\tau = 2.29 \times 10^{+1}$, the same behaviour is observed. Moreover, as shown in Fig. 7.6, the CPU time regarding the FSS has also increased. This was expected, given that the problem became more coupled, thus requiring more effort to be solved by the FSS. On the other hand, the CPU time spent by FIM remained practically unchanged. Likewise, the same pattern is seen in Fig. 7.7, where the coupling intensity was further escalated to $4.63 \times 10^{+1}$. Therefore, the speedup behaviour is not influenced by τ.

Figure 7.6 – Speedup and CPU time achieved in the Terzaghi's column problem. $\tau = 2.29 \times 10^{+1}$, $N_v = 10770$, $N_t = 200$.



Source: Own authorship.

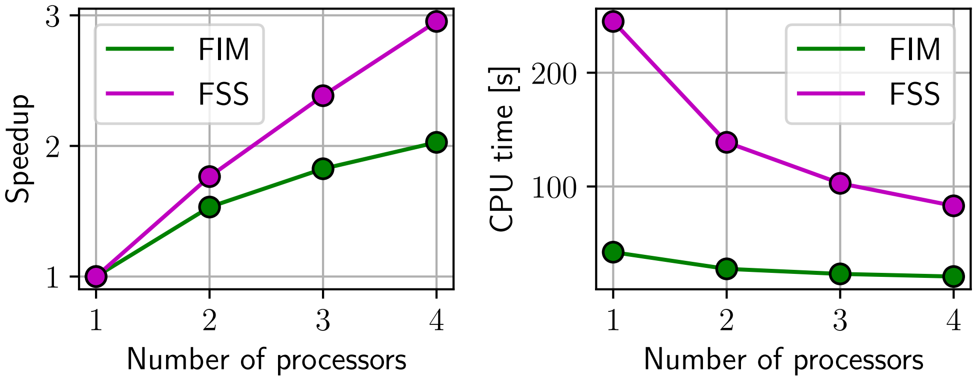
Figure 7.7 – Speedup and CPU time achieved in the Terzaghi's column problem. $\tau = 4.63 \times 10^{+1}$, $N_v = 10770$, $N_t = 200$.



Source: Own authorship.

In turn, the amount of time steps N_t concerning the simulation does not appear to impact the speedup pattern as well. Repeating the scenario in which the coupling intensity is 7.51×10^{-1}, and reducing N_t to 20, the speedup attained is pictured in Fig. 7.8, for the same 10770 control volume grid. In addition, the CPU time decrease has about the same order of the time step amount reduction.

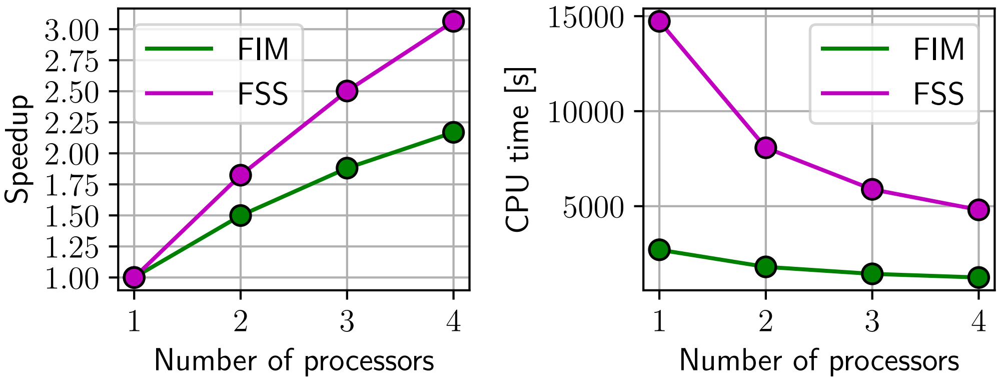
Figure 7.8 – Speedup and CPU time achieved in the Terzaghi's column problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 10770$, $N_t = 20$.



Source: Own authorship.

Furthermore, a more refined grid, with 54444 control volumes, is exploited. As a result, the FIM speedup is observed to grow, reaching 117% with 4 processors, while the FSS speedup stayed unaltered, as displayed in Fig. 7.9. However, as more processors are employed, the FSS looks more attractive, given that the difference between the two speedups still seems to increase in favour of the iterative coupling scheme. This was also observed in the previous results shown.

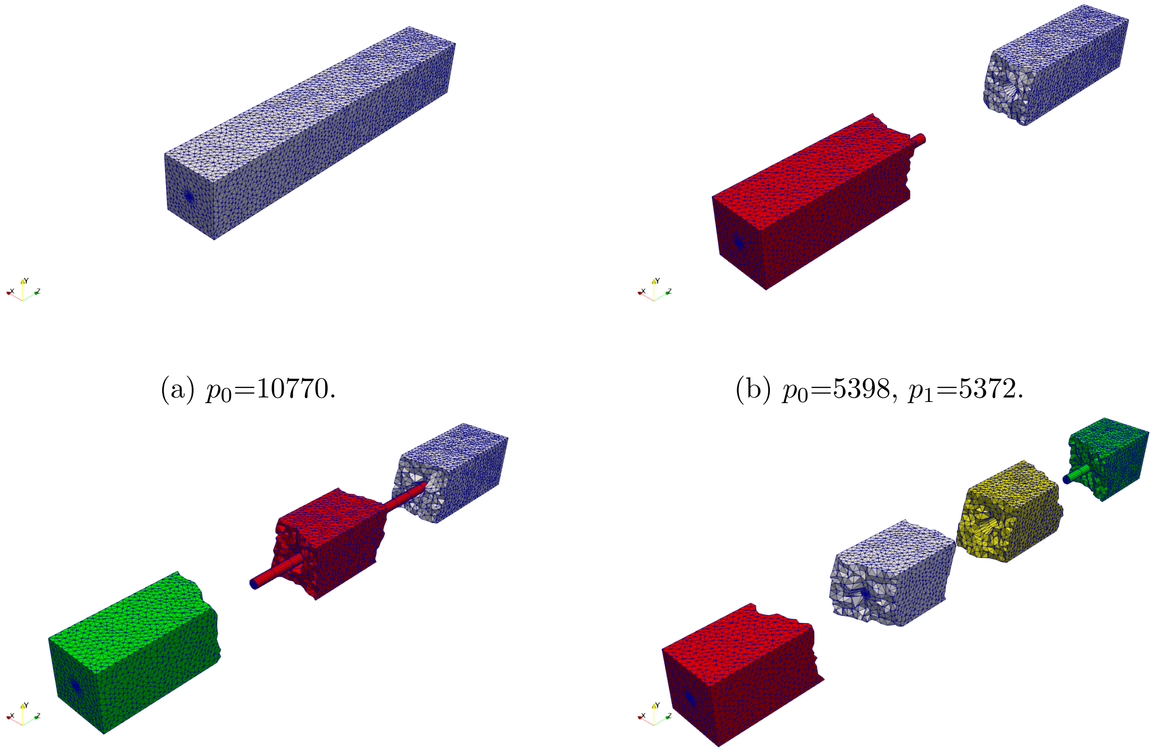
Figure 7.9 – Speedup and CPU time achieved in the Terzaghi's column problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 54444$, $N_t = 200$.



Source: Own authorship.

At last, Fig. 7.10 shows the partitions used in the Terzaghi's column simulations for different amount of processors. The maximum difference between the number of vertices assigned to each partition was never superior to 5%, which is quite satisfactory.

Figure 7.10 – Exploded view of the Terzaghi's column partitions, the subcaption indicates the number of vertices belonging to each partition p. Partitions 0, 1, 2 and 3 are painted in white, red, green and yellow, respectively.



(a) p_0=10770.

(b) p_0=5398, p_1=5372.

(c) p_0=3523, p_1=3599, p_2=3648.

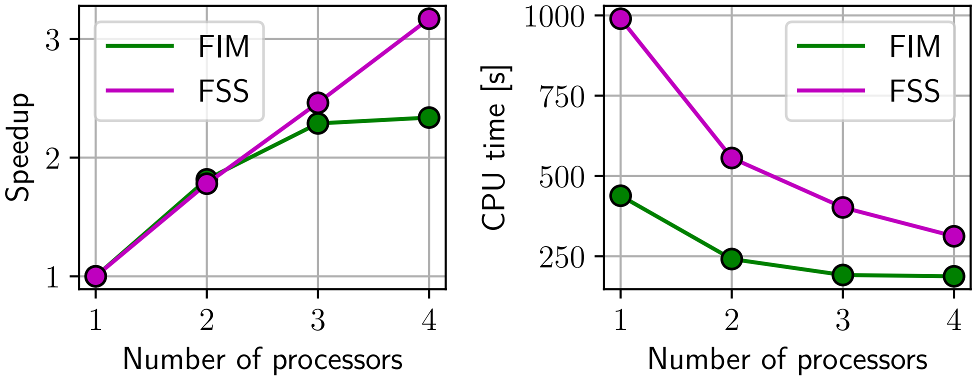
(d) p_0=2690, p_1=2765, p_2=2626, p_3=2689.

Source: Own authorship.

7.2.4 Cryer's sphere

The speedup achieved for the Cryer's sphere was somewhat larger than the speedup obtained for the Terzaghi's column, as shown in Fig. 7.11. However, the speedup followed the same pattern as the one displayed in the preceding problem, *i.e*, it presented the same values regardless of the coupling intensity or the amount of time steps simulated. Therefore, all speedups presented in this subsection were obtained for $\tau = 7.51 \times 10^{-1}$ and $N_t = 200$, being all the other cases exhibited in Annex A.

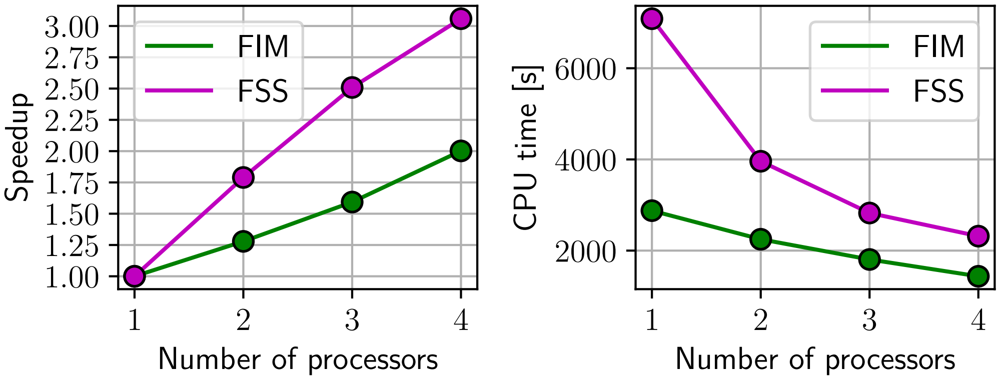
Figure 7.11 – Speedup and CPU time achieved in the Cryer's sphere problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 10060$, $N_t = 200$.



Source: Own authorship.

The only difference in the speedup behaviour appeared when the grid was refined, as happened in the Terzaghi's column. Nevertheless, for the Cryer's sphere, the speedup decreased for the refined computation domain, opposing the pattern observed in the previous subsection. This reduction is pictured in Fig. 7.12, where speedup decreased from 134% to 100% for the 54498 vertex grid.

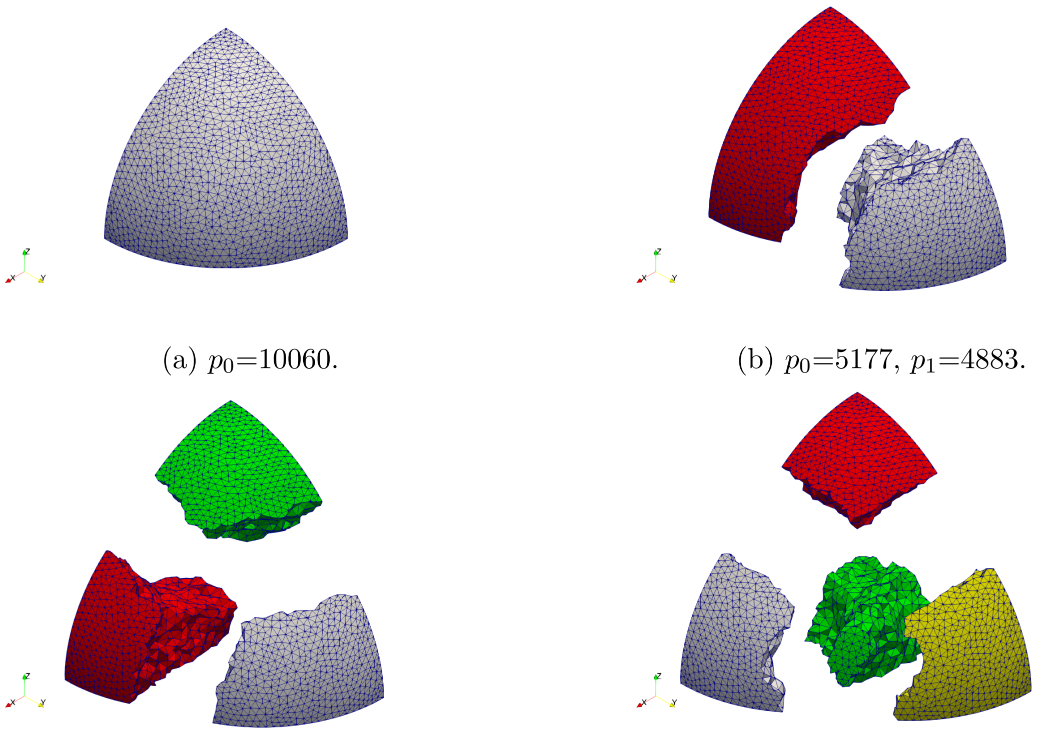
Figure 7.12 – Speedup and CPU time achieved in the Cryer's column problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 54498$, $N_t = 200$.



Source: Own authorship.

Finally, Fig. 7.13 shows the exploded view of the partitions for different numbers of processors.

Figure 7.13 – Exploded view of the Cryer's sphere partitions, the subcaption indicates the number of vertices belonging to each partition p. Partitions 0, 1, 2 and 3 are painted in white, red, green and yellow, respectively.



(a) p_0=10060.

(b) p_0=5177, p_1=4883.

(c) p_0=3327, p_1=3300, p_2=3433.

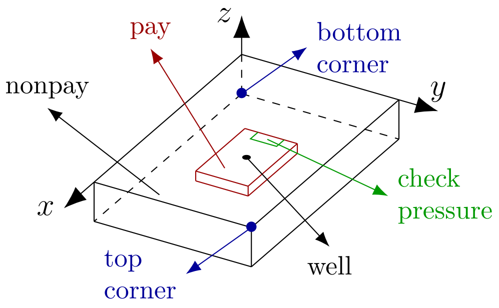
(d) p_0=2590, p_1=2512, p_2=2446, p_3=2512.

Source: Own authorship.

7.3 Reservoir depletion

The last problem was taken from Dean (DEAN et al., 2003) and it also a benchmark of the *Rede de Geomecânica*, the geomechanics research network of *Petrobras*. This scenario consists of soft reservoir, called *pay region*, that lays inside a stiff impermeable rock, the *nonpay region* — the nomenclature is due to the economic value of the fluid that lays within each porous medium. A vertical well is located in the centre of the reservoir, as portrayed in Fig. 7.14.

Figure 7.14 – Domain sketch.



Source: Own authorship.

Table 7.4 – Domain dimensions.

Region	Corners [km]
Nonpay	(0,000; 0,0000; -3,18516) (18,897; 9,4488; 0,0000)
Pay	(6,096; 3,048; -3,1242) (12,801; 6,400; -3,048)

Source: Own authorship.

Regarding the boundary conditions, all boundaries are sealed and have zero normal displacement, excluding the top boundary. Besides being free to deform, this surface has no initial stresses acting on it. Across the domain, there is a vertical stress gradient of $-1.1162 \times 10^{+4}$ Pa/m, being the gradients of the initial horizontal stresses equal to half of this value — these gradients represent total stresses. In turn, the initial pressure at each control volume is given by

$$(p_0)_i = \frac{\exp\left(z_i \rho_f c_f g_z\right) - 1}{c_f} + p_{surface} \tag{7.5}$$

with g_z being the z component of the gravity vector, whose value equals to -9,81 m^2/s. The initial pressure at the surface, $p_{surface}$, is equal to $1,01353 \times 10^{+05}$ Pa. At this same pressure, the reference fluid density ρ_f is $1.00 \times 10^{+3}$ kg/m^3.

Furthermore, the well has a wellbore radius of 0,0762 m and it produces at rate of 0,09 m$^3/s$ for 4000 days. Throughout the simulation, a time step size of $\Delta t = 20$ days is used. The poroelastic properties of the nonpay and pay regions are listed in Table 7.5, as the fluid properties are displayed on Table 7.6. The simulations were carried through the FIM, as it performed better than the FSS.

Even though the volume-weighted pressure in the pay region decreases with time, as seen in Fig. 7.15, there are positions within the reservoir in which the pressure increases. For example, the pressure over time in position $\boldsymbol{r} = [6.40080,\ 4.72440,\ -3.05562]^T$ km is shown in Fig. 7.16, where the Mandel-Cryer effect is captured again — this time being

Table 7.5 – Poroelastic properties.

Property	Unit	Pay	Nonpay
ν	–	2.50×10^{-01}	2.50×10^{-01}
G	Pa	$2.76 \times 10^{+07}$	$2.76 \times 10^{+09}$
c_s	Pa^{-1}	0.00	0.00
ϕ	–	2.50×10^{-01}	2.50×10^{-01}
ρ	$kg \cdot m^{-3}$	$2.70 \times 10^{+03}$	$2.70 \times 10^{+03}$
k_{xx}	m^2	9.87×10^{-14}	0.00
k_{yy}	m^2	9.87×10^{-14}	0.00
k_{zz}	m^2	9.87×10^{-15}	0.00

Source: (DEAN et al., 2003).

Table 7.6 – Fluid properties.

Property	Unit	Fluid
ρ	$kg \cdot m^{-3}$	$1.00 \times 10^{+3}$
μ	$Pa \cdot s$	1.00×10^{-3}
c_f	Pa^{-1}	4.35×10^{-10}

Source: (DEAN et al., 2003).

more evident than the one observed in Fig. 7.2. In order to obtain the pressure in this position, it was necessary to compute the average of four control volumes, given that there is no vertex in $\boldsymbol{r}$.

Figure 7.15 – Volume-weighted reservoir pressure.

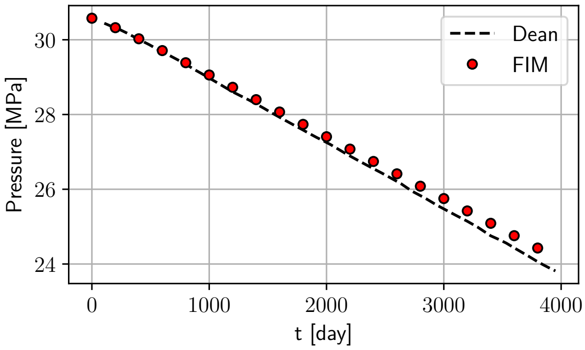
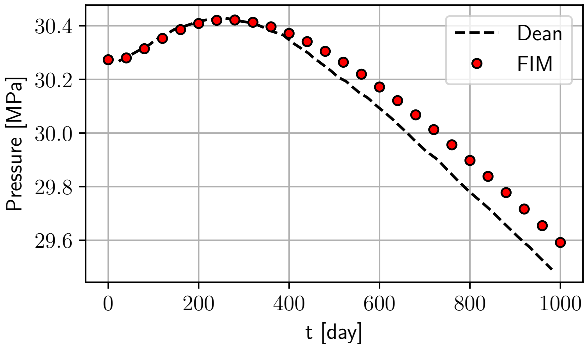


Figure 7.16 – Pressure on reservoir surface.



In turn, the subsidence at the well top and surface are shown in Fig. 7.17 and Fig. 7.18, respectively. As it may be observed, the vertical displacement at the well top is larger than at the surface. Therefore, the overburden expands, although it is under compression.

Figure 7.17 – Subsidence at the well top.

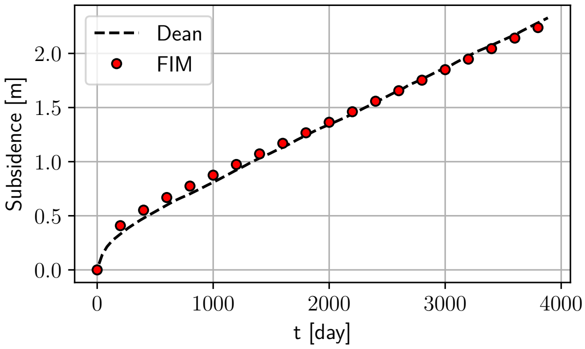
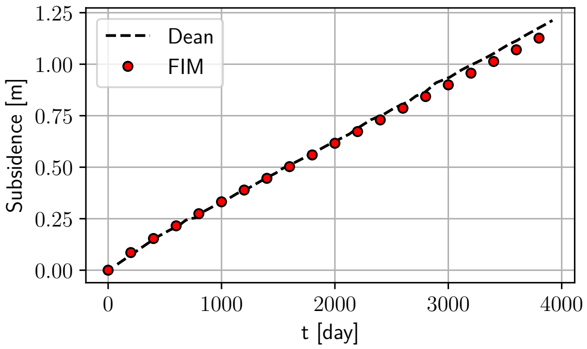
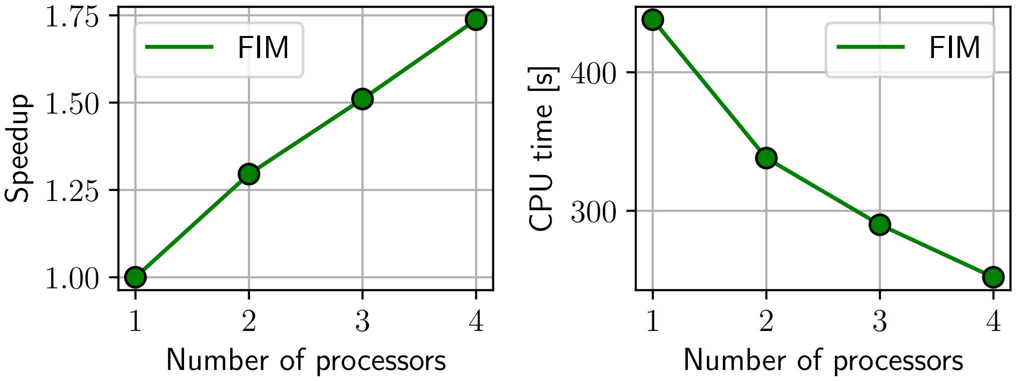


Figure 7.18 – Subsidence at the surface.



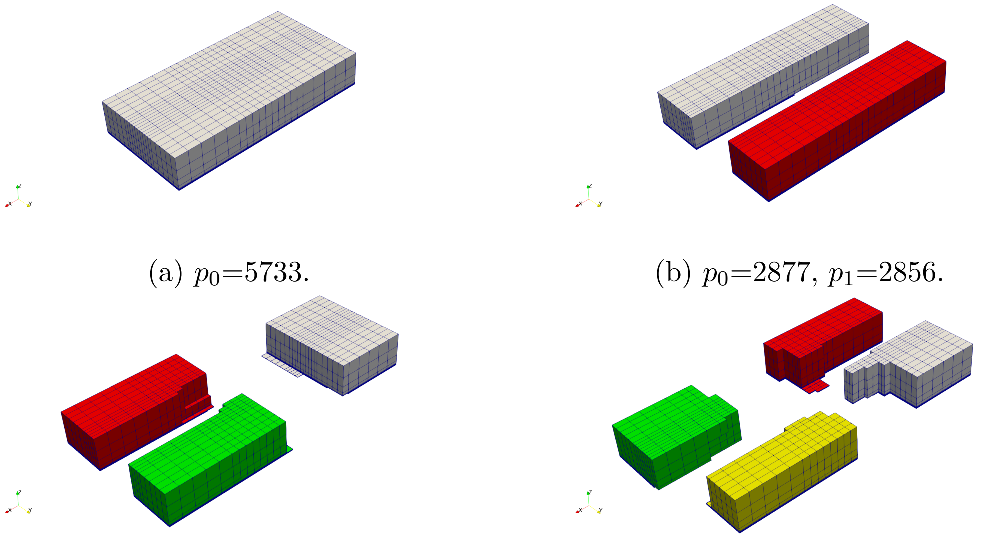
At last, Fig. 7.19 shows the speedup achieved for the reservoir depletion problem, as Fig. 7.20 pictures the exploded view of the partitions employed.

Figure 7.19 – Speedup and CPU time achieved in the reservoir depletion problem. $N_v = 5733$, $N_t = 200$.



Source: Own authorship.

Figure 7.20 – Exploded view of the reservoir depletion partitions, the subcaption indicates the number of vertices belonging to each partition p. Partitions 0, 1, 2 and 3 are painted in white, red, green and yellow, respectively.



(a) p_0=5733.

(b) p_0=2877, p_1=2856.

(c) p_0=1918, p_1=1942, p_2=1873.

(d) p_0=1429, p_1=1429, p_2=1426, p_3=1449.

Source: Own authorship.

8 Conclusion

The parallel performance of two poroelasticity coupling schemes, the Fully Implicit Method and the Fixed-Stress Split, has been compared. Even though the latter has shown a larger speedup, it is still outperformed by the former — this behaviour does not seem to be related to the coupling intensity and amount of time steps simulated. Also, the FSS speedup has shown to be independent of the grid refinement, appearing to be attached only to the boundary conditions and geometry of the domain.

On the other hand, the FIM speedup has shown two opposing behaviours with respect to the grid refinement. It increased for the Terzaghi's column, from 96% to 117%, and decreased for the Cryer's sphere, from 134% to 100% — in both cases, four processors have been employed. Since the linear solution is the most time-consuming task, concerning the FIM, this behaviour may be ascribed to the performance of the preconditioner and/or solver. It is important to remark that the preconditioning of coupled linear systems is not an easy task to accomplishment (VERDUGO; WALL, 2016), which makes its solution even more challenging.

In respect to the Domain Decomposition Method, the implemented algorithm yielded good quality partitions in all simulated scenarios. That is, the number of vertices attributed to each partition was roughly the same, and the number of ghost vertices created was satisfactorily small, given that the speedup due to the parallelization was larger than the inter-process communication cost — the CPU time always decreased as more processors were employed. Nevertheless, the partition quality can be improved even further by balancing the domain decomposition by constraints, which is achieved by assigning weights to the vertices and/or edges of the grid. Thus, when decomposing the domain, METIS would balance these weights among the partitions, besides following the two *criteria* discussed in Chap. 6. This would allow, for instance, to assign all domain wells to a single processor, therefore not requiring additional communication in the assemblage of Eq. 4.32.

8.1 Suggestions for future studies

The immediate suggestion would be a performance comparison between the FIM and FSS in the solution of nonlinear problems, such as two-phase flow coupled with poroelasticity or single-phase coupled with poroplasticity. These nonlinearities are often employed in commercial reservoir simulators due to its applicability in the oil and gas industry.

Furthermore, it would be interesting to verify the influence of the balanced domain decomposition in scenarios which concern prescribed flow wells, such as the reservoir depletion problem. This would require some investigation concerning the estimation of the vertices/edges weights in order to successfully decompose the domain.

At last, a study regarding the preconditioning and solving of coupled geomechanics linear systems is also advisable, given its impact in the total simulation time. In addition, it would be wise to employ open-source libraries in this research, such as PETSc, so to exploit its already available preconditioners and solvers.

Bibliography

BALAY, S. et al. *PETSc Web page*. 2019. <https://www.mcs.anl.gov/petsc>. Disponível em: <https://www.mcs.anl.gov/petsc>. Citado na página 50.

BIOT, M. A. General theory of three-dimensional consolidation. *Journal of applied physics*, AIP, v. 12, n. 2, p. 155–164, 1941. Citado 2 vezes nas páginas 19 and 26.

CHENG, A. H.-D. *Poroelasticity*. [S.l.]: Springer, 2016. Citado 2 vezes nas páginas 53 and 56.

DEAN, R. et al. A comparison of techniques for coupling porous flow and geomechanics. paper spe 79709. In: *SPE Reservoir Simulation Symposium*. [S.l.: s.n.], 2003. p. 3–5. Citado 2 vezes nas páginas 62 and 63.

DETOURNAY, E.; CHENG, A. H.-D. Fundamentals of poroelasticity. In: *Analysis and design methods*. [S.l.]: Elsevier, 1993. p. 113–171. Citado na página 27.

FANCELLO, E. A. *Natural and synthetic polymers – constitutive modelling and applications in biomechanics*. 2019. XL Ibero-Latin-American Congress on Computational Methods in Engineering, CILAMCE 2019. Citado na página 19.

GAMBOLATI, G.; FERRONATO, M.; JANNA, C. Preconditioners in computational geomechanics: a survey. *International Journal for Numerical and Analytical Methods in Geomechanics*, Wiley Online Library, v. 35, n. 9, p. 980–996, 2011. Citado na página 19.

GIACOMELLI, F. W. *DivideEtImpera: Domain decomposition for parallel numerical simulations*. 2019. <https://github.com/felipegiacomelli/DivideEtImpera/>. Disponível em: <https://github.com/felipegiacomelli/DivideEtImpera/>. Citado na página 49.

GREIN, E. A. *A Parallel Computing Approach Applied To Petroleum Reservoir Simulation*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2015. Citado na página 20.

HAUSER, T. Parallel I/O for the cgns system. In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. [S.l.: s.n.], 2004. p. 1088. Citado na página 31.

HONÓRIO, H. T. et al. A fully-conservative finite volume formulation for coupled poroelastic problems. In: *Proceedings of XXXIX Ibero-Latin American Congress on Computational Methods in Engineering, CILAMCE 2018*. [S.l.: s.n.], 2018. Citado na página 20.

HONÓRIO, H. T. et al. Counterintuitive physical behaviors in coupled consolidation problems. In: *Proceedings of XL Ibero-Latin-American Congress on Computational Methods in Engineering, CILAMCE 2019*. [S.l.: s.n.], 2019. Citado na página 52.

HONÓRIO, H. T. et al. A performance comparison between coupling schemes for poroelasticity. In: *Proceedings of the 25th ABCM International Congress of Mechanical Engineering, COBEM-2019*. [S.l.: s.n.], 2019. Citado 2 vezes nas páginas 20 and 54.

HURTADO, F. S. V. *Formulação tridimensional de volumes finitos para simulação de reservatórios de petróleo com malhas não-estruturadas híbridas.* Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2012. Citado na página 29.

KARYPIS, G.; KUMAR, V. *METIS-Serial graph partitioning and fill-reducing matrix ordering.* 2013. <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>. Disponível em: <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>. Citado na página 46.

KIM, J.; TCHELEPI, H. A.; JUANES, R. Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 200, n. 13-16, p. 1591–1606, 2011. Citado 3 vezes nas páginas 19, 42, and 54.

LI, X. S.; DEMMEL, J. W. Superlu_dist: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Transactions on Mathematical Software (TOMS)*, ACM, v. 29, n. 2, p. 110–140, 2003. Citado na página 55.

MANDEL, J. Consolidation des sols (étude mathématique). *Geotechnique*, Thomas Telford Ltd, v. 3, n. 7, p. 287–299, 1953. Citado na página 52.

MCTIGUE, D. Thermoelastic response of fluid-saturated porous rock. *Journal of Geophysical Research: Solid Earth*, Wiley Online Library, v. 91, n. B9, p. 9533–9542, 1986. Citado na página 26.

MURAD, M. *A new computational scheme for solving the black-oil model incorporating geomechanical coupling.* 2019. XL Ibero-Latin-American Congress on Computational Methods in Engineering, CILAMCE 2019. Citado na página 19.

PEACEMAN, D. W. et al. Interpretation of well-block pressures in numerical reservoir simulation (includes associated paper 6988). *Society of Petroleum Engineers Journal*, Society of Petroleum Engineers, v. 18, n. 03, p. 183–194, 1978. Citado na página 39.

POIRIER, D. et al. The cgns system. In: *29th AIAA, Fluid Dynamics Conference.* [S.l.: s.n.], 1998. p. 3007. Citado na página 31.

TERZAGHI, K. v. Die berechnung der durchlassigkeitsziffer des tones aus dem verlauf der hydrodynamischen spannungserscheinungen. *Sitzungsberichte der Akademie der Wissenschaften in Wien, Mathematisch-Naturwissenschaftliche Klasse, Abteilung IIa*, v. 132, n. 3–4, p. 125–138, 1923. Citado na página 23.

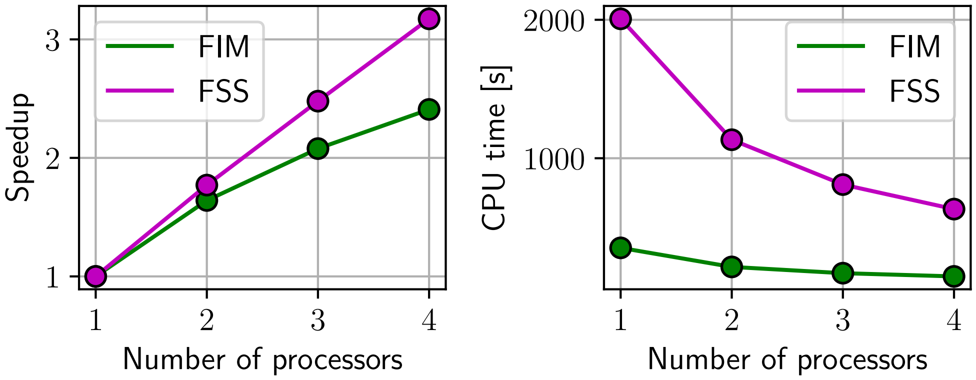
VERDUGO, F.; WALL, W. A. Unified computational framework for the efficient solution of n-field coupled problems with monolithic schemes. *Computer Methods in Applied Mechanics and Engineering*, Elsevier, v. 310, p. 335–366, 2016. Citado na página 65.

VERRUIJT, A. *Theory and problems of poroelasticity.* Netherlands: Delft University of Technology, 2016. Citado na página 54.

Annex

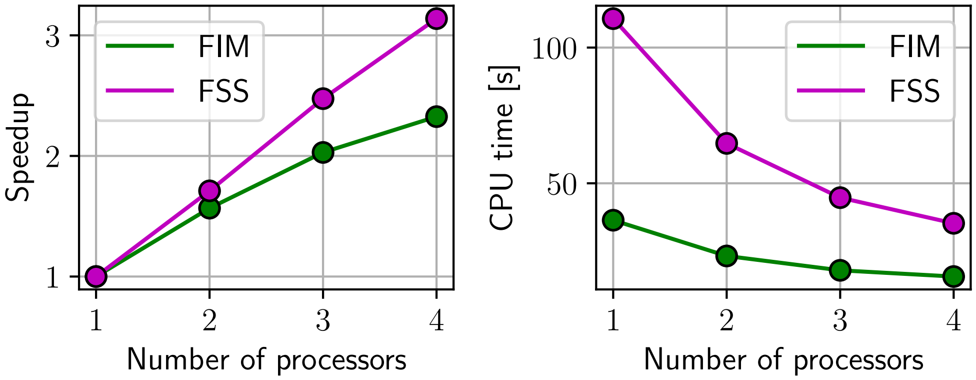
ANNEX A – Speedup comparison - Cryer's sphere

Figure A.1 – Speedup and CPU time achieved in the Cryer's sphere problem. $\tau = 4.63 \times 10^{+1}$, $N_v = 10060$, $N_t = 200$.



Source: Own authorship.

Figure A.2 – Speedup and CPU time achieved in the Cryer's sphere problem. $\tau = 7.51 \times 10^{-1}$, $N_v = 10770$, $N_t = 20$.



Source: Own authorship.