UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA CIVIL

FILIPE ANTÔNIO CUMARU SILVA ALVES

A MULTISCALE CONTROL VOLUME FRAMEWORK USING 3D UNSTRUCTURED GRIDS FOR THE SIMULATION OF SINGLE PHASE FLOW IN ANISOTROPIC AND HETEROGENEOUS POROUS MEDIA

Recife
2023

FILIPE ANTÔNIO CUMARU SILVA ALVES

A MULTISCALE CONTROL VOLUME FRAMEWORK USING 3D UNSTRUCTURED GRIDS FOR THE SIMULATION OF SINGLE PHASE FLOW IN ANISOTROPIC AND HETEROGENEOUS POROUS MEDIA

Master's thesis submitted to the Civil Engineering Graduate Program as a partial fulfilment requirement to obtain the title of Master of Science in Civil Engineering.

Program of study: Simulation and management of petroleum reservoirs

Supervisor: Paulo Roberto Maciel Lyra, PhD
Co-supervisor: Darlan Karlo Elisiário de Carvalho, PhD

Recife
2023

Catalogação na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

A474m Alves, Filipe Antônio Cumaru Silva.
A Multiscale Control Volume framework using 3D unstructured grids for the simulation of single phase flow in anisotropic and heterogeneous porous media / Filipe Antônio Cumaru Silva Alves – 2023.
59 f.: il., fig., tab. e siglas.

Orientador: Prof. Dr. Paulo Roberto Maciel Lyra.
Coorientador: Prof. Dr. Darlan Karlo Elisiário de Carvalho.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Civil, 2023.
Inclui Referências e Apêndices.
Texto em inglês.

1. Engenharia Civil. 2. Multiescala. 3. MsCV. 4. MPFA-D. 5. GLS. 6. Malha de fundo. 7. Escoamento monofásico. I. Lyra, Paulo Roberto Maciel (Orientador). II. Carvalho, Darlan Karlo Elisiário de (Coorientador). III. Título.

UFPE

624 CDD (22. ed.) BCTG/2023-175

FILIPE ANTÔNIO CUMARU SILVA ALVES

A MULTISCALE CONTROL VOLUME FRAMEWORK USING 3D UNSTRUCTURED GRIDS FOR THE SIMULATION OF SINGLE PHASE FLOW IN ANISOTROPIC AND HETEROGENEOUS POROUS MEDIA

Dissertação em Engenharia Civil da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito para obtenção do título de Mestre em Engenharia Civil, Área de Simulação e Gerenciamento de Reservatórios de Petróleo.

Aprovada em 24/07/2023

Orientador: Prof. Dr. Paulo Roberto Maciel Lyra – UFPE
Coorientador: Prof. Dr. Darlan Karlo Elisiário de Carvalho - UFPE

BANCA EXAMINADORA

participação por videoconferência
Prof. Dr. Ramiro Brito Willmersdorf (examinador interno)
Universidade Federal de Pernambuco

participação por videoconferência
Dr. Hermínio Tasinafo Honório (examinador externo)
Technical University of Delft

participação por videoconferência
Prof. Dr. Marcio Arab Murad (examinador externo)
Laboratório Nacional de Computação Científica

ACKNOWLEDGEMENTS

A scientific work is seldom the result of the effort of a single individual. One gets help on many fronts in order to produce a proper research, from the fellow researchers and advisors to funding agencies and people outside the academic environment.

First and foremost, I would like to thank Dr. Paulo Lyra and Dr. Darlan de Carvalho for advising me during the course of this Master's degree. Having guiding me since my Bachelor's degree, they truly honor the title of advisor by providing precious insights and enabling the development of the research of their students. I would also like to express my great gratitude to Dr. Artur de Souza whose work is not only the basis of this thesis, but who acted as an advisor and friend since I started as a research beginner student.

Second, I would like to thank all members of the PADMEC/LCCV research group. The quality of the work produced there has always inspired me to keep improving and to try to reach new heights. My special thanks to Dr. Ramiro Willmersdorf, Dr. Túlio Moura, Dr. José Cícero dos Santos, João Paulo Andrade and Maria Eduarda Galindo.

The development of this work was only possible through the funding and support of many agencies. For that, I would like to thank the Science Support Foundation of the State of Pernambuco (FACEPE), the National Council for Scientific and Technological Development (CNPq) and the Coordination for the Improvement of Higher Education Personnel (CAPES). It is important to acknowledge the work of the personnel from the Civil Engineering Graduate Program (PPGEC) and from the Institute for Petroleum and Energy (i-LITPEG).

Last but not least, I would like to thank my family for the unconditional support throughout this journey. My parents, Adalberto and Renata, my brother, João, my grandparents, Josué (Doda), Elza, Adalberto (Iu) and Célia, my uncle, Josué, and my aunt, Denise.

ABSTRACT

The level of detail on modern geological models requires higher resolution grids that may render the simulation of multiphase flow in porous media intractable. Moreover, these models may comprise highly heterogeneous media with phenomena taking place in different scales. The original Multiscale Finite Volume (MsFV) method can tackle such issues by constructing a set of numerical operators that map quantities from the fine-scale domain to a coarser one where the initial problem can be solved at a lower computational cost and the solution mapped back to the original scale. However, the MsFV formulation is limited to k-orthogonal grids since it uses a Two-point Flux Approximation (TPFA) method and employs an algorithm to generate the coarse meshes that is not capable of handling general geometries. The Multiscale Restriction Smoothed-Basis method (MsRSB) improves on the MsFV by introducing a new iterative procedure to find the multiscale operators and modifying the algorithm for the generation of the multiscale geometric entities to accommodate unstructured coarse grids, but is still limited to structured fine grids due to the TPFA discretization. Meanwhile, the Multiscale Control Volume method (MsCV) replaces the TPFA by the Multipoint Flux Approximation with a Diamond stencil (MPFA-D) scheme on the fine-scale while further enhancing the generation of the geometric entities to allow truly unstructured grids on the fine and coarse scales for two-dimensional simulation. In this work we propose an extension to three-dimensional geometries of both the MsCV and the algorithm to obtain the multiscale geometric entities based on the concept of background grid. We modify the MPFA-D to use the very robust Generalised Least Squares (GLS) interpolation technique to obtain the required auxiliary nodal unknowns. Finally, we also introduce an enhanced version of the 3-D MsCV with the incorporation of the enhanced MsRSB (E-MsRSB) to enforce M-matrix properties and improve convergence. We show that the 3-D MsCV method produces good results employing true unstructured grids on both scales to handle the simulation of the single-phase flow in anisotropic and heterogeneous porous media.

Keywords: multiscale; MsCV; MPFA-D; GLS; background grid; single-phase flow.

RESUMO

O nível de detalhe nos modelos geológicos modernos demanda o uso de malhas de alta resolução que podem tornar o problema da simulação do escoamento multifásico em meios porosos intratável. Além disso, estes modelos podem conter grande heterogeneidade e fenômenos que ocorrem em diferentes escalas. O Método dos Volumes Finitos Multiescala (MsFV) é capaz de lidar com tais problemas por meio da construção de um conjunto de operadores numéricos que mapeiam grandezas do domínio representado na escala de alta resolução para uma escala de menor resolução onde o problema inicial é resolvido a um custo computacional reduzido e cuja solução pode ser mapeada de volta à escala original. Contudo, a formulação do MsFV é limitada a malhas k-ortogonais devido ao uso do esquema de aproximação do fluxo por dois pontos (TPFA) e ao emprego de algoritmos para geração das malhas de menor resolução que não são capazes de tratar geometrias quaisquer. O método *Multiscale Restriction Smoothed-Basis* (MsRSB) melhora o MsFV introduzindo um novo procedimento iterativo para calcular os operadores multiescala e modificando o algoritmo de geração das entidades geométricas do multiescala para acomodar malhas de baixa resolução não-estruturadas, mas ainda é limitado ao uso de malhas estruturadas na escala de alta resolução pois mantém a discretização por TPFA. Enquanto isso, o *Multiscale Control Volume Method* (MsCV) substitui o TPFA pelo esquema *Multipoint Flux Appoximation with a Diamond Stencil* (MPFA-D) na escala de alta resolução e aprimora o procedimento para geração das entidades geométricas para permitir que malhas verdadeiramente não-estruturadas sejam usadas nas escalas de alta e baixa resolução para simulação numérica em duas dimensões. Neste trabalho, nós propomos uma extensão para geometrias tridimensionais do MsCV e do algoritmo para geração das entidades geométricas do multiescala baseado no conceito de malha de fundo. Nós também modificamos o MPFA-D para que seja usado o robusto método de interpolação das variáveis nodais *Global Least Squares* (GLS). Finalmente, introduzimos uma melhoria ao MsCV 3-D com a incorporação do método *Enhanced MsRSB* (E-MsRSB) para impor propriedades de uma matriz M à matriz MPFA-D e melhorar a convergência do método. Nós mostramos que o método MsCV 3-D produz bons resultados com o uso de malhas verdadeiramente não-estruturadas nas duas escalas para tratar a simulação do escoamento monofásico em meios porosos anisotrópicos e heterogêneos.

Palavras-chave: multiescala; MsCV; MPFA-D; GLS; malha de fundo; escoamento monofásico.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

TPFA	Two-Point Flux Approximation
MPFA-D	Multipoint Flux Approximation with a Diamond stencil
LS	Least Squares
LPC	Linearity Preserving Criterion
GLS	Global Least Squares
MsFV	Multiscale Finite Volume
MsRSB	Multiscale Restriction Smoothed Basis
E-MsRSB	Enhanced Multiscale Restriction-Smoothed Basis
MsCV	Multiscale Control Volume
E-MsCV	Enhanced Multiscale Control Volume

CONTENTS

1 INTRODUCTION

The numerical simulation of physical phenomena is a fundamental step to verify and predict how a proposed model will behave in a real world scenario. In the context of the Finite Volume formulations, this may involve discrete models whose resolution ranges from 10^8 to 10^9 (ZHOU, 2010; JARAMILLO et al., 2022). Moreover, the study can comprise phenomena happening in different scales and highly heterogeneous media, as it is often the case for the flow simulation in porous media (HAJIBEYGI et al., 2008). For these reasons, the computational complexity of the simulation can quickly render the problem intractable for conventional methods and hardware. For this purpose, scale transferring techniques are applied.

The scale transferring techniques can be divided into two main groups: the upscaling and the multiscale methods. The upscaling methods rely on a procedure, such as homogenization, to map the problem on the high resolution scale to a lower resolution where it is solved. These techniques are relatively simple and capable of effectively reducing the complexity of the initial problem. Classical local upscaling methods can produce good results on problems where the length scales of the heterogeneity are well separated (FARMER, 2002). Global upscaling methods can better handle more challenging scenarios such as the representation of near-well regions in the context of reservoir simulation (DURLOFSKY, 2005). Conversely, the multiscale methods compute a set of numerical operators to map quantities from the fine scale onto a coarser scale, and vice-versa. However, unlike upscaling, these operators allow to reconstruct an approximate solution on the fine-scale from the coarse-scale solution (CORTINOVIS, 2016).

The core idea behind each multiscale method is computation of the basis functions, i.e., the functions that form the scale-transferring numerical operators. In the Multiscale Finite Element method (MFEM) for elliptical problems, HOU; WU (1997) define local boundary conditions for each coarse block in order to find the basis functions. This approach uncouples the coarse blocks leading to a non-conservative velocity field. CHEN; HOU (2003) addressed the issues on the MFEM by introducing the Mixed Multiscale Finite Element method (MMFEM) which reimposes conservation on the velocity field by creating basis functions that compute the pressure and the velocity fields simultaneously. Furthermore, ARBOGAST; BRYANT (2002) adapted the MMFEM to the simulation of two-phase flows in petroleum reservoirs.

Among the proposed multiscale formulations, we turn our attention to the Multiscale Finite Volume (MsFV) method introduced by JENNY; LEE; TCHELEPI (2003) in the context of subsurface flow simulation. In this approach, an auxiliary dual coarse grid is employed in order to define the operators by solving local problems in this grid.

Many improvements have since been proposed to the MsFV such as the introduction of more complex physics to the model (LUNATI; JENNY, 2006; LEE; WOLFSTEINER; TCHELEPI, 2008) and an algebraic formulation of the MsFV (ZHOU; TCHELEPI, 2008; ZHOU; TCHELEPI, 2011).

Most of the formulations in the MsFV family are restricted to structured grids on both the fine and the coarse scales. This is mainly due to the usage of a Two-point Flux Approximation scheme (TPFA) on the fine-scale, which is only consistent on k-orthogonal grids, and the difficulty of generating the multiscale entities in unstructured geometries (SOUZA et al., 2020). In this sense, MØYNER; LIE 2016 proposed the Multiscale Restriction-Smoothed Basis (MsRSB) method to enable the use of unstructured coarse-scale grids. In their work, the authors present an iterative formulation of the multiscale operators, reducing the amount of recalculations during the simulation to modify the operators. However, the MsRSB still relies on the TPFA for the discretization on the fine-scale. This issue is partially addressed by BOSMA et al. (2021) who introduce an extension of the MsRSB for non M-matrices, the enhanced MsRSB (E-MsRSB). SOUZA et al. (2020) presented the Multiscale Control Volume (MsCV) which improves on the original MsRSB by replacing the TPFA with the Multipoint Flux Approximation with a diamond stencil (MPFA-D) scheme from GAO; WU (2011) and CONTRERAS et al. (2016), hence allowing for the use of truly unstructured grids on both scales for the simulation of the 2-D two-phase flow in porous media.

The MPFA-D requires the interpolation of the nodal unknowns involved in its flux expression. The choice of interpolation strategy is crucial and deeply affects the convergence of the method. In its 3-D formulation, as described by LIRA FILHO et al. 2021, the Linearity-Preserving Explicit Weight (LPEW3) interpolation is adopted. Although it observes the Linearity Preserving Criterion (LPC), it fails to compute accurate solutions in the presence of strong anisotropy. To overcome these limitations, DONG; KANG (2022) introduced the Global Least Squares (GLS) interpolation. As it is shown by CAVALCANTE (2023), it presents a more stable behaviour in the presence of strong anisotropy and highly heterogeneous media while still observing the LPC.

In the present thesis, we propose an extension of the MsCV (SOUZA et al., 2020) to 3-D geometries coupled with the 3-D MPFA-D (LIRA FILHO et al., 2021) and the robust GLS interpolation (DONG; KANG, 2022). In order to generate the multiscale geometric entities, we also extend to 3-D geometries the background grid framework proposed by SOUZA et al. (2022), so far restricted to 2-D models. Finally, we introduce an enhanced version of the 3-D MsCV, the E-MsCV, by incorporating the preconditioning technique from the E-MsRSB (BOSMA et al., 2021) to the definition of the multiscale operators. We apply the developed framework to the study of the single-phase flow in anisotropic and heterogeneous porous media.

1.1 Research objectives

The main objective of this thesis is to study and develop a 3-D multiscale framework for the simulation of single-phase incompressible flow in anisotropic and heterogeneous media.

1.1.1 Specific objectives

- Extend the Multiscale Control Volume method (MsCV) to 3-D models;
- Develop the 3-D extension of the multiscale pre-processing algorithm based on a background grid;
- Incorporate the robust GLS interpolation to the 3-D MPFA-D;
- Introduce the E-MsRSB in the MsCV framework to improve the convergence rate of the method.

1.2 Thesis organization

This thesis is structured as follows. This first chapter provides an introduction and a review of multiscale methods for the simulation of flows in porous media. The second chapter is dedicated to present the mathematical formulation used in this work. In the third chapter, we introduce the numerical formulation used for the discretization of the problem in the fine-scale. Next, we detail the 3-D multiscale formulation developed. In the fifth chapter, we describe the extension of the algorithm proposed for the generation of the multiscale geometric entities based on the concept of a background grid. The sixth chapter contains a series of examples devised to show the application of the multiscale framework applied to the simulation of single-phase flow in porous media. Finally, in the seventh and final chapter we present the conclusions and suggest future works. We also present a discussion on the implementation issues in the Appendix A.

2 MATHEMATICAL FORMULATION

In this chapter, we briefly present the governing equations and respective adopted assumptions for the 3-D single-phase flow of an incompressible fluid in anisotropic and heterogeneous porous media. From these equations, the partial differential equation (PDE) that dictates this phenomenon can be deduced.

2.1 Mass conservation equation

The mass conservation equation on porous media for the multi-phase flow over a physical domain $\Omega \times [0, t]$, with $\Omega \subset \mathbb{R}^3$, is given by (CARVALHO, 2005):

$$-\nabla \cdot (\rho_i \overrightarrow{v}_i) + q_i = \frac{\partial (\phi \rho_i S_i)}{\partial t}, \tag{2.1}$$

where ϕ is the porosity of the medium and $\overrightarrow{v}_i$, ρ_i, q_i and S_i correspond to the velocity, density, source/sink terms, and saturation of the phase i, respectively.

Let a single-phase flow of an incompressible fluid in a non-deformable medium. Neglecting any chemical or thermal reactions and the effects of adsorption and dispersion, then Equation (2.1) can be simplified as:

$$-\rho \nabla \cdot \overrightarrow{v} + q = 0 \quad \therefore \quad \nabla \cdot \overrightarrow{v} = Q \tag{2.2}$$

where $Q = q/\rho$. Equation (2.2) is the Poisson's equation for a diffusive phenomenon in three dimensions.

2.2 Darcy's law

First elucidated by Henry Darcy in 1856, Darcy's law is an empirical law that describes the flow of a fluid in a porous media under the following assumptions (EWING, 1983):

1. The fluid is considered to be a newtonian fluid;
2. The fluid does not chemically react with the medium;
3. The flow is considered to be laminar;

4. The permeability does not depend on the pressure and the temperature of the fluid;
5. The slippage effect of gases at low pressure is neglected;
6. Any electrokinetic effect is disregarded.

Hence, in the multiphase flow context, Darcy's law can be written as:

$$\overrightarrow{v}_i = -\underset{\sim}{\mathcal{K}}\lambda_i \left(\nabla p_i - \rho_i \overrightarrow{g}\right) \text{ with } \lambda_i = \frac{k_{ri}}{\mu_i}, \tag{2.3}$$

where λ_i, p_i, k_{ri} and μ_i stand for the mobility, the pressure, the relative permeability and the viscosity of the phase i, respectively, $\overrightarrow{g}$ represents the gravitational effect, and $\underset{\sim}{\mathcal{K}}$ is a positive definite symmetric full permeability tensor satisfying the ellipticity condition (BORSUK; KONDRATIEV, 2006; VÉRON, 2004).

The absolute permeability $\underset{\sim}{\mathcal{K}}$ is a measure of the capacity of the medium to allow a fluid to flow. In turn, the relative permeability is a dimensionless parameter that gauges the effective permeability of the phase. The permeability tensor can be expressed in Cartesian coordinates as follows:

$$\underset{\sim}{\mathcal{K}}(\overrightarrow{x}) = \begin{pmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{xy} & K_{yy} & K_{yz} \\ K_{xz} & K_{yz} & K_{zz} \end{pmatrix}. \tag{2.4}$$

Once again assuming a single-phase flow of an incompressible fluid in a rigid medium and neglecting the effects of gravity, Equation (2.3) can be simplified as:

$$\overrightarrow{v} = -\underset{\sim}{\mathcal{K}}\nabla p. \tag{2.5}$$

2.3 Elliptic pressure equation

Given equations (2.2) and (2.5), we can write the elliptic pressure equation as:

$$\nabla \cdot \overrightarrow{v} = Q, \text{with } \overrightarrow{v} = -\underset{\sim}{\mathcal{K}}\nabla p. \tag{2.6}$$

Typical boundary conditions for Equation (2.6) are given by:

$$p = g_D \quad \text{for } \overrightarrow{x} \in \Gamma_D, \tag{2.7}$$

$$\overrightarrow{v} \cdot \overrightarrow{n} = g_N \quad \text{for } \overrightarrow{x} \in \Gamma_N, \tag{2.8}$$

where Γ_D and Γ_N represent the Dirichlet and Neumann boundaries, respectively, and $\overrightarrow{n}$ denotes the unit outward normal vector. The global domain boundary is given by $\partial\Omega = \Gamma_D \cup \Gamma_N$, such that $\Gamma_D \cap \Gamma_N = \varnothing$

3 NUMERICAL FORMULATION

In this chapter, we discuss the finite volume discretization used for the elliptic pressure equation (2.6) at the fine-scale. Here, we summarize the main aspects of the MPFA-D scheme and the GLS interpolation for the vertex unknowns.

3.1 Finite volume formulation

To discretize Equation (2.6) at the fine-scale, we have employed the Multipoint Flux Approximation with a Diamond stencil (MPFA-D) method (LIRA FILHO et al., 2021), replacing the original nodal interpolation algorithm with the Generalized Least Squares (GLS) interpolation of the vertex unknowns (DONG; KANG, 2022).

To ensure clarity, we adopt an overloaded notation and consider Ω to also represent the discrete computational domain, while Γ denotes its associated boundary. Additionally, Ω can be subdivided into a set of n_k non-overlapping control volumes. By integrating Equation (2.6) over an individual control volume $\hat{R}$ and applying Gauss's theorem, the following expression is obtained:

$$\int_{\Omega_{\hat{R}}} \nabla \cdot \overrightarrow{v} \, \mathrm{d}\Omega_{\hat{R}} = \int_{\Gamma_{\hat{R}}} \overrightarrow{v} \cdot \overrightarrow{n} \, \mathrm{d}\Gamma_{\hat{R}} = \int_{\Omega_{\hat{R}}} \mathcal{Q} \, \mathrm{d}\Omega_{\hat{R}}, \tag{3.1}$$

where $\Omega_{\hat{R}}$ and $\Gamma_{\hat{R}}$ denote the volume and the boundary of the control volume $\hat{R}$, respectively. By applying the mean value theorem, Equation (3.1) can be rewritten as:

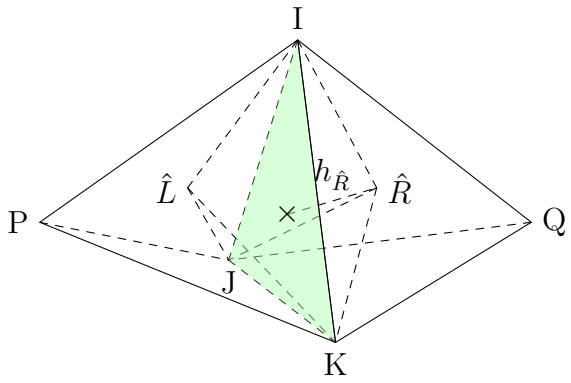
$$\sum_{m \in \Gamma_{\hat{R}}} (\overrightarrow{v} \cdot \overrightarrow{N})|_m = \overline{\mathcal{Q}}_{\hat{R}} \Omega_{\hat{R}}, \tag{3.2}$$

where $\overline{\mathcal{Q}}_{\hat{R}}$ is the average source term in $\hat{R}$, m is a face of $\hat{R}$ and $\overrightarrow{n}$ is the area vector associated to m. As discussed in (AAVATSMARK et al., 1998a; AAVATSMARK et al., 1998b), various strategies can be employed to approximate the flux expression described in Equation (3.2), leading to the development of different schemes.

3.2 The Multipoint Flux Approximation with a Diamond stencil (MPFA-D)

The MPFA-D provides a full pressure support (EDWARDS; ZHENG, 2008) discretization of Equation (2.6) for 3-D tetrahedral meshes. Given the arrangement shown in Figure 1, the flux through the face IJK is approximated by:

Figure 1 – Two tetrahedra $\hat{R}$ and $\hat{L}$ sharing a face IJK illustrating the main entities in the MPFA-D scheme.



Source: (LIRA FILHO et al., 2021).

$$\overrightarrow{v}_{\hat{R}} \cdot \overrightarrow{N}_{IJK} \approx -K^n_{eff}\left[2(p_{\hat{R}} - p_{\hat{L}}) - D_{JI}(p_I - p_J) - D_{JK}(p_K - p_J)\right], \tag{3.3}$$

where K^n_{eff} is the transmissibility coefficient given by a harmonic average of the orthogonal projection of the permeability tensor onto the face IJK written as:

$$K^n_{eff} = \frac{K^n_{\hat{R}} K^n_{\hat{L}}}{K^n_{\hat{R}} h_{\hat{L}} + K^n_{\hat{L}} h_{\hat{R}}}, \tag{3.4}$$

and D_{JI} and D_{JK} are the cross-diffusion terms given by:

$$D_{JI} = \frac{\overrightarrow{\tau_{JK}} \cdot \overrightarrow{\hat{L}\hat{R}}}{|\overrightarrow{N}_{IJK}|^2} - \frac{1}{|\overrightarrow{N}_{IJK}|}\left(\frac{K^{JK}_{\hat{R}}}{K^n_{\hat{R}}} h_{\hat{R}} + \frac{K^{JK}_{\hat{L}}}{K^n_{\hat{L}}} h_{\hat{L}}\right) \tag{3.5}$$

$$D_{JK} = \frac{\overrightarrow{\tau_{JI}} \cdot \overrightarrow{\hat{L}\hat{R}}}{|\overrightarrow{N}_{IJK}|^2} - \frac{1}{|\overrightarrow{N}_{IJK}|}\left(\frac{K^{JI}_{\hat{R}}}{K^n_{\hat{R}}} h_{\hat{R}} + \frac{K^{JI}_{\hat{L}}}{K^n_{\hat{L}}} h_{\hat{L}}\right), \tag{3.6}$$

with:

$$K^n_{\hat{r}} = \frac{\overrightarrow{N}^T_{IJK} \underset{\sim}{K}_{\hat{r}} \overrightarrow{N}_{IJK}}{2|\overrightarrow{N}_{IJK}|} \tag{3.7}$$

$$K^{ij}_{\hat{r}} = \frac{\overrightarrow{N}^T_{IJK} \underset{\sim}{K}_{\hat{r}} \overrightarrow{\tau}_{ij}}{2|\overrightarrow{N}_{IJK}|} \tag{3.8}$$

$$\overrightarrow{\tau}_{ij} = \overrightarrow{N}_{IJK} \times \overrightarrow{ij}, \tag{3.9}$$

for i,$j = I, J, K$ and $r = R, L$.

3.2.1 Boundary conditions treatment

The flux on a face submitted to Dirichlet boundary conditions is approximated by:

$$\overrightarrow{v}_{\hat{R}} \cdot \overrightarrow{N}_{IJK} \approx -\left[2\frac{K^n_{\hat{R}}}{h_{\hat{R}}}(p_{\hat{R}} - g^D_J) + D_{JI}(g^D_J - g^D_I) + D_{JK}(g^D_J - g^D_K)\right], \tag{3.10}$$

where g^D_I, g^D_J and g^D_K are the prescribed values on the boundary, with D_{JI} and D_{JK}:

$$D_{JI} = \frac{\left(\overrightarrow{\tau_{JK}} \cdot \overrightarrow{J\hat{R}}\right)}{|\overrightarrow{N}_{IJK}|}\frac{K^{(n)}_{\hat{R}}}{h_{\hat{R}}} + K^{JK}_{\hat{R}}, \tag{3.11}$$

$$D_{JK} = \frac{\left(\overrightarrow{\tau_{JI}} \cdot \overrightarrow{J\hat{R}}\right)}{|\overrightarrow{N}_{IJK}|}\frac{K^{(n)}_{\hat{R}}}{h_{\hat{R}}} + K^{JI}_{\hat{R}}. \tag{3.12}$$

Furthermore, for faces on the Neumann boundary, the boundary condition value is:

$$\overrightarrow{v}_{\hat{R}} \cdot \overrightarrow{N}_{IJK} = g_N. \tag{3.13}$$

3.2.2 Vertex unknowns interpolation

As it can be seen from Equation (3.3), the MPFA-D's unique flux expression, apart from the cell unknowns, includes vertex unknowns that must be eliminated in order to obtain a completely cell-centered approximation. This can be achieved by rewriting the vertex variables as a linear combination of the surrounding cell-centered values:

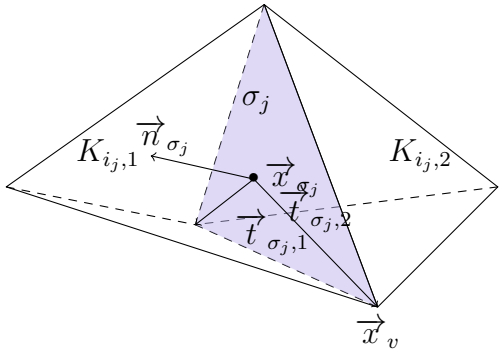
$$p_v = \sum_{\hat{k}=1}^{n_k} \omega_{\hat{k}} p_{\hat{k}}. \tag{3.14}$$

Here, we have opted to use the Global Least Squares (GLS) interpolation (DONG; KANG, 2022). It is a linear-preserving interpolation technique capable of handling heterogeneous and highly anisotropic media while maintaining a good convergence rate as discussed by CAVALCANTE (2023).

DONG; KANG (2022) introduce the following metric for the magnitude of the anisotropy of the permeability coefficient $\underset{\sim}{\mathcal{K}}$:

$$\mathcal{A}(\underset{\sim}{\mathcal{K}}) = \left(1 - \frac{3(\det \underset{\sim}{\mathcal{K}})^{1/3}}{\operatorname{tr} \underset{\sim}{\mathcal{K}}}\right)^2. \tag{3.15}$$

Figure 2 – Local structure and notation for an internal node.



Source: (DONG; KANG, 2022).

It allows to take into account the physical aspects of the problem in addition to the geometric ones during the calculation of the weights.

A piecewise linear function is also defined around the interpolated node v:

$$P_i(\overrightarrow{x}) = \overrightarrow{g}_i^T(\overrightarrow{x} - \overrightarrow{x}_v) + p_v, \tag{3.16}$$

in which $\overrightarrow{g}_i$ are the coefficients of the linear combination and $\overrightarrow{x}_v$ is the position of the node v.

For an internal node, as illustrated in Figure 2, the weights of the GLS interpolation are computed by finding the least squares (LS) solution of:

$$\min_{\mathbf{V}} \left(\sum_{i=1}^{n_K} (\delta U_i)^2 + \sum_{j=1}^{n_f} \left[(\delta F_j)^2 + (\delta T_{j,1})^2 + \tau_{j,2}^2 (\delta T_{j,2})^2 \right] \right), \tag{3.17}$$

where n_K and n_f are the number of control volumes and the number of faces surrounding node v, respectively. Furthermore:

$$\tau_{j,2} = |\overrightarrow{t}_{\sigma_j,2}|^{-\eta_j} \tag{3.18}$$

$$\eta_j = \max\left(\mathcal{A}(\underset{\sim}{\mathcal{K}}_{i_{j,1}}), \mathcal{A}(\underset{\sim}{\mathcal{K}}_{i_{j,2}})\right) \tag{3.19}$$

$$\begin{aligned}
\delta U_i &= P_i(\overrightarrow{x}) - p_i, \\
\delta F_j &= \overrightarrow{n}_{\sigma_j} \underset{\sim}{\mathcal{K}}_{i_{j,1}} \overrightarrow{g}_{i_{j,1}} - \overrightarrow{n}_{\sigma_j} \underset{\sim}{\mathcal{K}}_{i_{j,2}} \overrightarrow{g}_{i_{j,2}}, \\
\delta T_{j,1} &= \overrightarrow{t}^T_{\sigma_j,1} \overrightarrow{g}_{i_{j,1}} - \overrightarrow{t}^T_{\sigma_j,1} \overrightarrow{g}_{i_{j,2}}, \\
\delta T_{j,2} &= \overrightarrow{t}^T_{\sigma_j,2} \overrightarrow{g}_{i_{j,1}} - \overrightarrow{t}^T_{\sigma_j,2} \overrightarrow{g}_{i_{j,2}}, \\
\mathbf{U} &= (p_1, \ldots, p_{n_k})^T, \\
\mathbf{V} &= (\overrightarrow{g}^T_1, \ldots, \overrightarrow{g}^T_{n_k}, p_v)^T,
\end{aligned} \tag{3.20}$$

and $\overrightarrow{t}_{\sigma_j,1}$, $\overrightarrow{t}_{\sigma_j,2}$ are non co-linear tangent vectors to the face σ_j.

The problem described by Equation (3.17) can be rewritten as:

$$\mathbb{M}_v \mathbf{V} = \mathbb{N}_v \mathbf{U}, \tag{3.21}$$

whose LS solution is:

$$\mathbf{V} = \left(\mathbb{M}_v^T \mathbb{M}_v\right)^{-1} \mathbb{M}_v^T \mathbb{N}_v \mathbf{U}. \tag{3.22}$$

It follows that:

$$(\omega_1, \ldots, \omega_{n_k}) = \overrightarrow{e} \left(\mathbb{M}_v^T \mathbb{M}_v\right)^{-1} \mathbb{M}_v^T \mathbb{N}_v, \tag{3.23}$$

where $\overrightarrow{e}$ is the last column of a $(3n_k + 1) \times (3n_k + 1)$ identity matrix, i.e., the weights correspond to the last row of $\left(\mathbb{M}_v^T \mathbb{M}_v\right)^{-1} \mathbb{M}_v^T \mathbb{N}_v$.

For nodes on the Neumann boundary, the interpolated value takes the form:

$$p_v = \sum_{\hat{k}=1}^{n_k} \omega_{\hat{k}} p_{\hat{k}} + \omega_c, \tag{3.24}$$

in which ω_c stands for the contribution of the Neumann boundary value to p_v.

The minimization problem seen in Equation (3.17) is modified to include the contribution of the boundary condition, so that the new problem to be solved in the LS sense is:

$$\min_{\mathbf{V}} \left(\sum_{i=1}^{n_K} (\delta U_i)^2 + \sum_{j=1}^{n_f} \left[(\delta F_j)^2 + (\delta T_{j,1})^2 + \tau_{j,2}^2 (\delta T_{j,2})^2 \right] + \sum_{k=1}^{n_b} (\delta N_k)^2 \right), \tag{3.25}$$

where:

$$\delta N_k = -\overrightarrow{n}_{\sigma_{k,b}}^T \underset{\sim}{\mathcal{K}}_{i_k} \overrightarrow{g}_{i_k} - g_N \left(\overrightarrow{x}_{\sigma_{k,b}} \right), \tag{3.26}$$

n_b is the number of boundary faces surrounding the node and $g_N \left(\overrightarrow{x}_{\sigma_{k,b}} \right)$ corresponds to the Neumann boundary condition value at face $\sigma_{k,b}$.

The final solution for the weights can be found similarly to the internal nodes. For a more detailed description of the assembly procedure of these local problems, refer to (DONG; KANG, 2022).

4 MULTISCALE FORMULATION

4.1 Multiscale framework

The core idea behind every multiscale scheme is to use the coarse scale as an auxiliary basis that can be used to approximate the solution of the fine-scale system of equations. This is achieved by using two numerical operators that, together, are capable of projecting information back and forth between these two discrete scales: the prolongation operator $\mathbf{P}$ and the restriction operator $\mathbf{R}$. The prolongation operator stores the basis functions, that in turn, capture the influence that each coarse volume has on its corresponding support region. On the other hand, the restriction operator $\mathbf{R}$ maps the distribution of the fine-scale quantities onto the coarse-scale. In practical terms, for a problem with a fine-scale and coarse-scale containing n_f, and n_c control volumes, respectively, $\mathbf{P}$ and $\mathbf{R}$ are matrices with dimensions $n_f \times n_c$ and $n_c \times n_f$.

By definition $\mathbf{P}$ approximates the fine-scale solution $\mathbf{p}$ by projecting the coarse scale solution $\mathbf{p}_c$ onto the fine-scale space, as it follows:

$$\mathbf{p} \approx \mathbf{p}_{ms} = \mathbf{P}\mathbf{p}_c. \tag{4.1}$$

where $\mathbf{p}_{ms}$ denotes the multiscale approximate solution.

Let the fine-scale discrete system of equations be:

$$\mathbf{A}\mathbf{p} = \mathbf{q}. \tag{4.2}$$

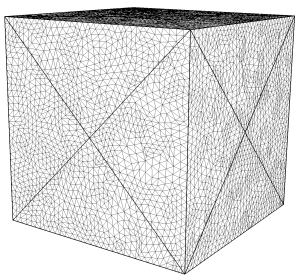
In order to find $\mathbf{p_c}$, we need to find a system defined on the coarse scale, similarly to Equation (4.2). By replacing the approximation of the exact solution of Equation (4.1) in (4.2) and premultiplying by the restriction operator, a coarse-scale system is found:

$$\mathbf{RA}\left(\mathbf{P}\mathbf{p}_c\right) = \mathbf{R}\mathbf{q} \quad \therefore \quad \mathbf{A}_c\mathbf{p}_c = \mathbf{q}_c, \tag{4.3}$$

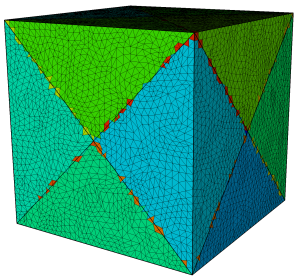
where $\mathbf{A}_c = \mathbf{RAP}$ and $\mathbf{q}_c = \mathbf{Rq}$.

Multiscale methods differentiate themselves based on how the prolongation and restriction operators are defined. In this thesis, we extend the Multiscale Control Volume (MsCV) (SOUZA et al., 2020) to 3-D domains. The MsCV method uses the prolongation operator of the Multiscale Restriction-Smoothed Basis (MsRSB) (MØYNER; LIE, 2016) in combination with the MPFA-D in 2-D. The resulting framework allows the use of

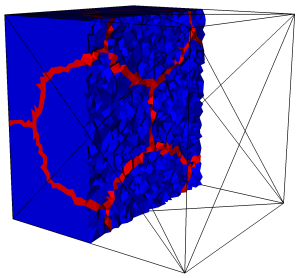
Figure 3 – Illustration of the MsCV geometric entities.



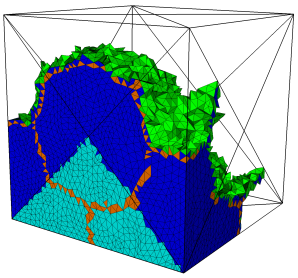
(a) The fine-scale mesh Ω_f (thin lines) and the background grid Ω_{bg} (bold lines).



(b) The primal coarse mesh Ω_c^P.



(c) The dual coarse mesh Ω_c^D (red).



(d) An example of support region I_j (dark blue) and its boundary B_j (green) highlighting H_j (orange) and the corresponding primal volume (light blue).

Source: Author.

unstructured meshes on all scales. To adapt the MsCV to general 3-D grids, two issues need to be addressed: the use of a consistent flux approximation and the definition of an algorithm capable of generating the multiscale geometric entities on these grids. The first issue is resolved by replacing the standard 2-D MPFA-D by the 3-D MPFA-D (LIRA FILHO et al., 2021) in which we introduce the robust GLS interpolation as defined on the previous chapter. The latter is addressed by extending the background grid strategy presented by SOUZA et al. (2022) to 3-D geometries. Furthermore, we also propose an enhanced version of the 3-D MsCV, the E-MsCV, which applies the preconditioning technique described by BOSMA et al. (2021) to improve convergence.

4.2 Multiscale geometric entities

In this section, we will summarize the geometric entities employed by the MsCV and the MsRSB methods based on the concepts introduced by (SOUZA et al., 2020; SOUZA et al., 2022; MØYNER; LIE, 2016). The algorithms used to generate these entities will be discussed in Chapter 5. Illustrations to the concepts presented below are provided in Figure 3.

Fine-scale mesh (Ω_f) The fine-scale mesh is the higher resolution discretization of the physical domain. It is usually the same grid used to estimate the physical properties of the medium.

Background grid (Ω_{bg}) The background grid is an auxiliary grid used as a reference to partition the fine-scale mesh and generate the primal and dual coarse meshes. This is a concept introduced in the multiscale context by SOUZA et al. (2022) for the 2-D case. Figure 3a presents both the fine-scale and the background grid.

Primal coarse mesh (Ω_c^P) This is a lower-resolution grid obtained by agglomerating volumes from the fine-scale mesh, as it can be seen in Figure 3b. In the original MsCV and MsRSB methods, this mesh is generated via some partitioning tool. However, in this work we use the concept of a background grid to determine the multiscale entities.

Dual coarse mesh (Ω_c^D) The dual coarse mesh is an auxiliary grid used to enforce mass conservation on the boundaries of the primal coarse volumes and later used in the flux reconstruction algorithm to find a conservative flux field from the multiscale solution. This grid is represented in Figure 3c by the red regions.

Primal coarse center (x^P) The primal coarse center is the fine-scale volume closest to the centroid of the corresponding primal coarse volume. MØYNER; LIE (2016) and BARBOSA et al. (2018) discuss different approaches to define the primal coarse center, but we have chosen to employ the aforementioned definition for the sake of simplicity.

Support region of a primal coarse volume j (I_j) The support region can be interpreted as the region of influence of a primal coarse center in the global domain. This is equivalent to:

$$\mathbf{P}_{i,j} \neq 0, \quad \forall i \in I_j. \tag{4.4}$$

It is important to note that the primal coarse center itself, as well as the support boundary B_j, is not part of the support region.

Support boundary of a primal coarse volume j (B_j) The support boundary consists of all cells that share at least one face with a cell in the support region I_j but are not a part of it themselves.

Global support boundary (G) The global support boundary is the set of all cells that belong to the support boundary of a primal coarse volume, i.e.:

$$G = \bigcup_{j=1}^{n_c} B_j \tag{4.5}$$

Global support boundary in a support region (H_j) This is the set of cells that are in the global support boundary and belong to the support region of j. Equivalently:

$$H_j = I_j \cap G \tag{4.6}$$

Figure 3d illustrates an example of a support region, its boundary and the intersection with H_j.

4.3 The MsCV operators in 3-D

The MsCV method proposed by SOUZA et al. (2020) is based on the MsRSB (MØYNER; LIE, 2016) which uses an iterative process to define the basis functions for the prolongation operator. We use the foundations of the original MsCV with some adaptations for the 3-D case to define the iterative procedure of constructing the prolongation operator.

The prolongation operator is initialized as the characteristic function of each primal coarse volume, i.e.,

$$P_{ij}^0 = \begin{cases} 1 & \text{if } \Omega_{f,i} \in \Omega_{c,i}^P \\ 0 & \text{otherwise} \end{cases}. \tag{4.7}$$

As pointed out by MØYNER; LIE (2016), other initial guesses could be used, but this choice is made for its simplicity and because it already provides partition of unit. The initial operator is then modified through weighted Jacobi iterations of the form:

$$P_j^{n+1} = P_j^n - \overline{\omega} D^{-1} A^{pre} P_j^n, \tag{4.8}$$

where $\overline{\omega}$ is the relaxation parameter of the Jacobi iteration set to $2/3$, D^{-1} is the inverse of the main diagonal of the preconditioned MPFA-D left-hand side term, and A^{pre} is the preconditioned MPFA-D matrix. Here, the preconditioned matrix is a direct application of the technique described by SOUZA et al. (2020) and is given by:

$$A_{ij}^{pre} = \begin{cases} A_{ij} & \text{if } i \neq j \\ A_{ii} - \sum_{k=1}^{n_f} A_{ik} & \text{otherwise} \end{cases}. \tag{4.9}$$

Given the iterative process in Equation (4.8), the smoothing procedure applied to compute each basis function P_j is detailed in Algorithm 1. The fourth step of the procedure is performed in order to ensure mass conservation since the fine-scale cells involved in this

calculation contribute only to the support region of the coarse cell j and, therefore, should hold the maximum value in the prolongation operator. Figure 4 illustrates the smoothing procedure of the basis functions.

Algorithm 1: The MsCV iterative smoothing procedure of the basis functions

Input: The initial guess $\mathbf{P}^0$

Output: The prolongation operator $\mathbf{P}$

Compute the initial increment $\hat{d}_j = -\overline{\omega} D^{-1} A^{pre} P_j^n$;

Modify $\hat{d}_j$ to ensure the partition of unity and avoid growth outside the support region:

$$d_{ij} = \begin{cases} \frac{\hat{d}_{ij} - P_{ij}^n \sum_{k \in H_j} \hat{d}_{ik}}{1 + \sum_{k \in H_j} \hat{d}_{ik}} & \text{if } \Omega^{f,i} \in H_j \\ \hat{d}_{ij} & \text{if } \Omega^{f,i} \in I_j \text{ and } \Omega^{f,i} \notin H_j \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

Set $P_j^{n+1} = P_j^n + d_j$;

Set $P_{i,j}^{n+1} = 1$ for all fine-scale cells i that belong solely to the support region of the primal center j;

Rescale P_j to ensure partition of unity, i.e., set $P_j^{n+1} = P_j^{n+1} / \sum_i P_{i,j}$;

Calculate the local error for cells outside the global support boundary:

$$e_j = \max_{i \notin G} |d_{ij}| \tag{4.11}$$

If $\|\mathbf{e}\|_\infty \leq tol$, stop and set $P = P^{n+1}$. Else, go to step 1;

Finally, for the restriction operator, we use the Finite Volume restriction operator described by JENNY; LEE; TCHELEPI (2003) defined as:

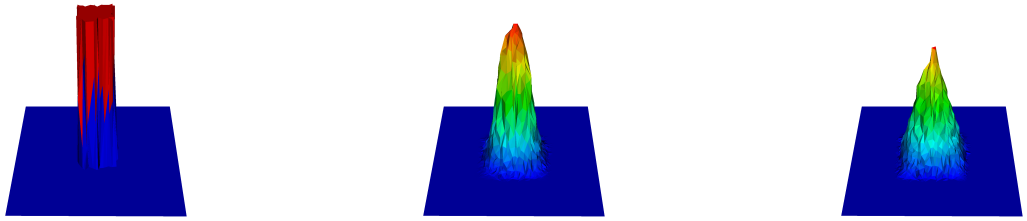
$$\mathbf{R}_{ij} = \begin{cases} 1 & \text{if } \Omega^{f,j} \in \Omega_{c,i}^P \\ 0 & \text{otherwise} \end{cases}. \tag{4.12}$$

4.4 The Enhanced MsCV (E-MsCV)

As discussed by BOSMA et al. (2021), the MsRSB's prolongation operator may show slow convergence when applied to non M-matrices. To overcome this issue, a modification to the fine-scale matrix is suggested so that the M-matrix properties are reinforced, making the convergence rate of the method closer to when it is applied to a TPFA matrix.

As we will show in our experiments, the 3-D MsCV presents the same convergence issues. Since the MsCV prolongation operator is based on the MsRSB, it is natural to

Figure 4 – The smoothing process of the multiscale basis functions.



(a) Initial guess. (b) 100 iterations. (c) 500 iterations.

Source: Author.

consider the application of the aforementioned procedure to our new framework. We designate this modified version of the MsCV the *Enhanced MsCV* (E-MsCV).

The E-MsCV improves on the MsCV by adopting the preconditioning technique proposed by BOSMA et al. (2021). It modifies the transmissibility matrix A by filtering all positive off-diagonal entries. This ensures M-matrix properties and improves the convergence rate of the Jacobi iterations. This preconditioning technique can be written as:

$$A^*_{ij} = \min\left(A_{ij}, 0\right), \text{ for } i \neq j, \tag{4.13}$$

$$A^{pre}_{ij} = \begin{cases} A^*_{ij} & \text{if } i \neq j \\ A^*_{ii} - \sum_{k=1}^{n_f} A^*_{ik} & \text{otherwise} \end{cases}. \tag{4.14}$$

The remaining of the E-MsCV follows the same iterative procedure described in Algorithm 1.

4.5 Flux reconstruction algorithm

By construction, the multiscale solution is mass conservative on the coarse scale. However, if the prolongated solution is used to calculate the fluxes on the fine scale, the resulting field will not be mass conservative since the algorithm used to obtain the prolongation operator decouples the domain for the solution of local problems within each support region. Hence, one must compute a new pressure field to accommodate the error introduced by the initial multiscale solution while keeping the mass conservation on the interfaces of the primal coarse cells. In this work, a procedure based on JENNY; LEE; TCHELEPI (2003) and SOUZA et al. (2020) is employed to obtain such pressure field.

The reconstructed pressure field is composed by the original multiscale solution on the boundaries of the primal coarse volume and a new solution computed in the interior

of the coarse cell. First, we determine the fluxes on the surface of each primal coarse cell using the prolongated solution. Next, for each coarse cell, we use the fluxes calculated on the previous step as Neumann boundary conditions to solve Equation (2.6) restricted to the primal coarse volume. This is can be written as:

$$\begin{cases} -\nabla \cdot (\underset{\sim}{K}\nabla \overline{p}) = q_f & \text{in } \Omega^P, \\ \nabla \overline{p} \cdot \overrightarrow{n} = v^{ms} & \text{on } \partial\Omega^P, \\ \overline{p} = p_{ms} & \text{in} x^P \text{if } \partial\Omega^P \cap \Gamma_D = \varnothing \end{cases}, \tag{4.15}$$

where $\overline{p}$ is the new pressure field computed inside each primal coarse cell, $\overrightarrow{n}$ is the outward normal unit vector to the interface, p_{ms} is the initial multiscale solution obtained by the prolongation of the coarse scale solution and v^{ms} is the flux field computed using p_{ms}. The latter condition on Equation (4.15) ensures that each local problem is well posed even for the primal coarse volumes that do not contain any volumes on the fine-scale on the Dirichlet boundary by forcing the primal coarse center to hold its initial value.

5 THE BACKGROUND GRID FRAMEWORK

One of the key phases of any variant of the MsFV framework is the generation of the primal and dual coarse grids. Properly defining these meshes can substantially reduce the number of steps taken by the iterative multiscale procedure.

A common approach to generate the primal coarse grid is by subdividing the computational domain using a partitioning tool such as Metis (KARYPIS; KUMAR, 1998). That is the alternative adopted by (MØYNER; LIE, 2016; BOSMA et al., 2017; SOUZA et al., 2020). Although straightforward, this procedure has the shortcoming of often being solely based on the geometry of the domain which in turn may lead to inconsistencies and loss of accuracy on the solution in the presence of highly heterogeneous media, as pointed out by MØYNER; LIE (2016) and MEHRDOOST (2019), who resort to mesh adaptation to mitigate these issues.

SOUZA et al. (2022) introduce a new procedure to generate the multiscale coarse grids by employing an auxiliary *background grid* that guides the definition of both the primal and dual coarse meshes. The background grid scheme intends to follow three assumptions:

1. The fine-scale grid is derived from the geological discretization of the domain;
2. The primal coarse grid should conform to the geological features of the medium; and
3. As far as possible, the grids from the background grid framework should be appliable to the upscaling context. Hence, strongly non-convex volumes should be avoided.

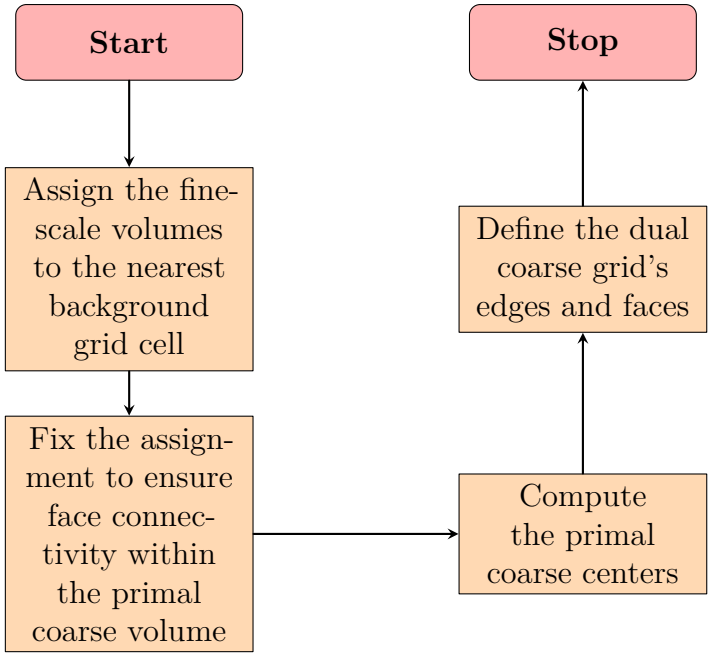
It is worth pointing out that the fine-scale and the background grid are independent from each other. The grids may have different topologies and the background grid is not necessarily a coarser version of the fine-scale grid. In fact, the background grid may be adapted to features in the physical domain and not follow the profile of the fine-scale discretization.

In this chapter, we present a novel 3-D extension of the original 2-D background grid framework and apply it to generate the multiscale coarse grids used for the simulations. The main steps of the algorithm are outlined by the flowchart in Fig. 5.

5.1 Primal coarse grid generation

The first step of the background grid framework is the generation of the primal coarse grid's entities, namely the volumes and faces. This procedure is described in

Figure 5 – Main steps of the background grid framework for 3-D geometries.



Source: Author.

Algorithm 2.

Firstly, it is necessary to find the fine cells owned by each background grid volume. The ownership relation is defined by checking if the background grid volume contains the fine cell's centroid. Assuming that the background grid is formed only by convex polyhedra, this verification is performed by constructing a Delaunay tessellation (DELAUNAY et al., 1934) and checking if the point is within one of the simplexes.

Next, we need to guarantee that the clusters of fine cells generated are connected by face, that is, if a fine cell in a cluster shares at least a face with another fine cell in the same cluster. This is done iteratively by checking the fine volumes in a disconnected cluster and reassigning them to a new primal coarse volume so that now the face connectivity criterion is obeyed. The *while* loop in Algorithm 2 performs this step. By making those checks iteratively, we ensure that the status of a fine volume that could not be reassigned is not reevaluated until all the other fine cells have been checked as well.

After all corrections are performed, we can define the primal coarse faces. For each primal coarse volume assembled, we retrieve the fine-scale faces that are in this primal volume and filter those that either are in the domain's boundary or are shared with another primal coarse cell. At the end, we have the primal coarse faces, i.e., the surface that delimits the corresponding primal coarse cell.

Algorithm 2: Primal coarse grid generation

Input: The fine-scale grid and the background grid
Output: The primal coarse grid
Assign each fine cell to the background grid volume that contains its centroid;
For each cluster of fine cells, construct a graph of face connectivity within the set, i.e., an edge in the graph indicates that a face is shared between two fine volumes;
Set S as a linked list containing the fine volumes in disconnected components of the graphs from the previous step;
while *S is not empty* **do**
    Get the first volume from S;
    if *the volume is connected to a well formed primal coarse volume* **then**
        Assign the volume to this primal volume;
    else
        Push the volume to end of S;
    end
end
for *each primal coarse volume C_j assembled* **do**
    $F_j \leftarrow$ the set of fine-scale faces forming the fine cells in C_j;
        $F_j^1 \leftarrow$ the fine faces in F_j that are in the domain's boundary;
        $F_j^2 \leftarrow$ the fine faces in F_j that are shared with other primal coarse cells;
    $\partial C_j \leftarrow F_j^1 \bigcup F_j^2$, the primal coarse faces of C_j;
end

5.2 Primal coarse centers calculation

Given the primal coarse grid, we can now determine the primal coarse centers of each primal coarse volume. As discussed in Section 4.2, the primal coarse centers are the fine cells whose centroids are the closest to the centroid of the background grid volume corresponding to the primal coarse volume. Similarly, we can define the primal coarse faces centers as the fine-scale face whose centroid is the nearest to the centroid of the corresponding background grid face. Both definitions are detailed in Algorithm 3.

5.3 Dual coarse grid generation

Once we have defined the primal coarse grid, the next step of the background grid framework is to generate the dual coarse grid. Here, we take the approaches devised by SOUZA et al. (2020) and SOUZA et al. (2022) as the basis for our own procedure to delimit the dual coarse grid.

Algorithm 3: Primal coarse centers computation

Input: The fine-scale grid, the background grid and the primal coarse grid
Output: A mapping of the centers for each primal coarse volume
for *each primal coarse volume* C_j **do**
  $D^j \leftarrow$ the euclidean distances between the fine cells in C_j and the centroid of the background grid volume corresponding to C_j;
  $x_j^P \leftarrow \arg\min_{k \in C_j} D_k^j$;
end
for *each primal coarse face* F_j **do**
  $D^j \leftarrow$ the euclidean distances between the fine faces in F_j and the centroid of the background grid face corresponding to F_j;
  $x_j^F \leftarrow \arg\min_{k \in F_j} D_k^j$;
end

5.3.1 Dual coarse edges definition

A dual grid edge can be defined as the shortest path between a primal coarse center and the center of a primal coarse face in the same volume. Since we are using the background grid as reference, we can interpret the shortest path geometrically, i.e., as the line segment connecting both points. Then, the dual coarse edge can be delimited as the set of fine-scale volumes intercepted by this line segment.

Algorithm 4 summarizes the procedure described above. The line segment-fine mesh intersection is computed by checking if the line segment intersects any of the fine faces that form a fine volume, which reduces the problem to computing a line-plane intersection. Furthermore, by adopting this strategy, we can guarantee the face connectivity within the dual coarse edge since if the line segment intercepts an internal fine face, then it intercepts both fine volumes sharing this face.

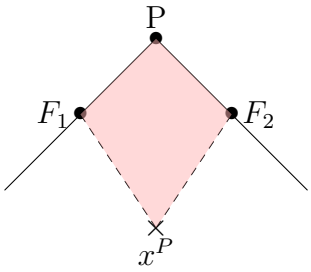
Algorithm 4: Dual coarse edges definition

Input: The fine-scale grid, the background grid and the primal coarse grid
Output: An assignment of fine-scale volumes to dual coarse edges
$\mathcal{E} \leftarrow$ an empty dictionary to store the fine volumes in each dual coarse edge;
for *each primal coarse volume* C_j **do**
  $x^{C_j} \leftarrow$ the centroid of C_j's center;
  for *each face of* C_j **do**
    $x^{F_j} \leftarrow$ the centroid of the face's center;
    $L \leftarrow$ the line segment connecting x^{C_j} and x^{F_j};
    $V \leftarrow$ the set of fine volumes in C_j that are intercepted by L;
    $\mathcal{E}\left[(C_j, F_j)\right] \leftarrow V$;
  end
end

5.3.2 Dual coarse faces definition

The dual coarse faces are the remaining entities in the dual coarse grid to be defined. Similar to the previous stages of the pre-processing algorithm, the background grid is used as a reference and stand-in for the primal coarse grid. For each background grid volume, we establish that a dual face is delimited by the quadrilateral formed by the centroid of the volume, the centroids of two neighboring faces and the point of intersection between the plane formed by the previous points and the edge shared by the chosen faces. Figure 6 provides a 2-D view of this arrangement where x^P is the centroid of the background grid volume, F_1 and F_2 are the centroids of two adjacent faces, and P is the point of intersection between the plane section $\overline{x^P F_1 F_2}$ and the edge shared between the adjacent faces.

Figure 6 – The arrangement of the points that define a dual face from a 2-D perspective. All points lying inside the red region form a dual face.



Source: Author.

Algorithm 5: Dual coarse faces definition

Input: The fine-scale grid, the background grid and the primal coarse grid
Output: An assignment of fine-scale volumes to dual coarse faces
$\mathcal{D} \leftarrow$ an empty dictionary to store the fine volumes in each dual coarse face;
for *each background grid volume* C_j^{bg} **do**
  $F_j^{adj} \leftarrow$ all pairs of adjacent faces that form C_j^{bg};
  $N \leftarrow$ the set of the normal vectors to the planes defined by the centroids of C_j^{bg} and each pair of faces in F_j^{adj};
  for *each plane* P_k *defined in the previous step* **do**
    Retrieve the fine volumes inside C_j^{bg} that are intercepted by the plane;
    $\mathcal{D}_k \leftarrow$ the fine volumes that are intercepted by P_k and whose centroids' projection is inside the region delimited by the points that form P_k;
    $\mathcal{D}\left[\left(C_j^{bg}, F_{j,k}^{adj}\right)\right] \leftarrow \mathcal{D}_k$, where $F_{j,k}^{adj}$ is the pair of adjacent faces whose centroids form P_k;
  end
end

The outer loop in Algorithm 5 determines all pairs of adjacent faces in the background grid volume, i.e., all faces that share an edge, and then computes the planes defined

by the centroid of the background grid volume and the centroids of each pair of faces. Once those are well defined, we can proceed to check which fine cells are intercepted by a plane and lie within the region of the dual face. The first step is done by simply checking if a cell has vertices in opposing sides of the plane.

In order to filter the fine volumes that are inside the region of the dual face, we verify if the projection of the cell's centroid is inside the dual face. Although the plane section of the dual face is formed by the three points of the plane P_k plus the point of intersection between the plane and the edge shared by the adjacent faces, we need only to check if the projection sits between the two line segments that connect the background grid volume centroid to the adjacent faces. If so, then the fine cell is part of the dual face. At the end, a dictionary of all fine volumes in a dual face is assembled.

5.4 Construction of the support regions

Once the multiscale coarse grids are defined, we can pass on to construct the support regions of each primal coarse volume. In the present approach, we take advantage of the dual coarse grid to delimit the boundaries of the support regions so that we do not have to compute new surfaces as it is done in the original MsCV pre-processing algorithm (SOUZA et al., 2020).

The steps to construct the support region of each primal coarse volume are outlined in the Algorithm 6. At the outer *for* loop, we define the primal coarse volumes involved in the support region of C_j. Next, using the background grid, we can establish the bounding surface of this cluster formed by dual faces around the primal coarse volume of interest. This bounding surface is used as a preliminary support boundary since it already contains all fine cells that actually form B_j but it also includes additional cells that belong to the support region I_j. An example of this initial support boundary is shown in Figure 7 for the structured case.

Given the initial support boundary, the support region itself can be now defined. In our procedure, a growth strategy is applied by adding fine volumes to I_j until the preliminary support boundary is reached. This idea is similar to a breadth first search in graphs. In this case, our target nodes would be the fine cells in B_j^* and a branch in the search algorithm would stop as soon it reaches one of those volumes. Finally, we fix the support boundary so that the additional cells are removed and included in the support region.

Algorithm 6: Construction of the support regions

Input: The fine-scale grid, the primal coarse grid, the dual coarse grid and the background grid

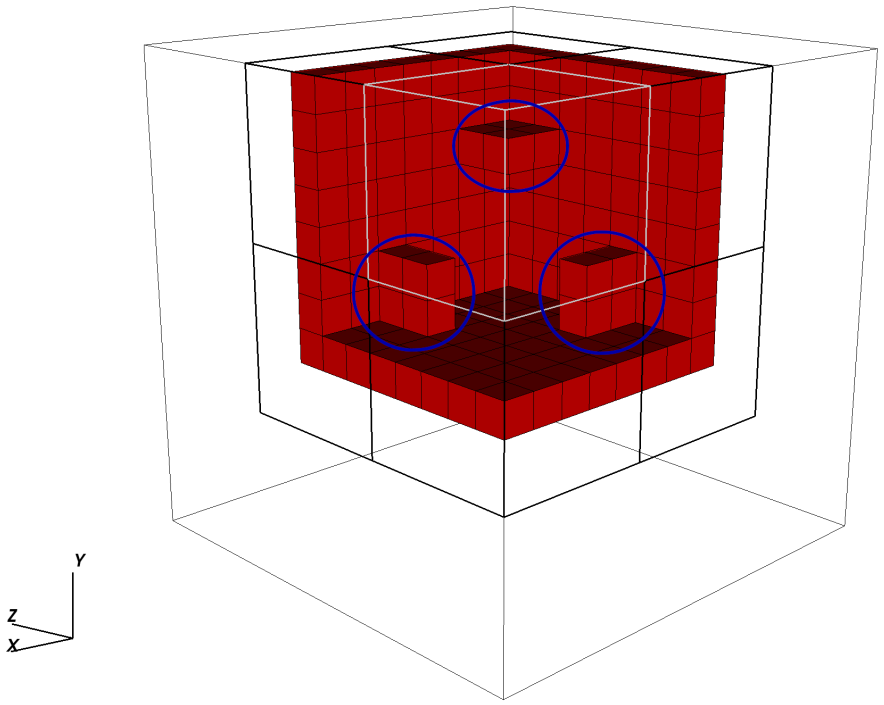
Output: The support region of each primal coarse center

for *each primal coarse volume* C_j **do**

- Find the primal volumes sharing at least a node with C_j using the background grid as reference;
- Get the background grid faces from the cells in the previous step that form the boundary of this set;
- Retrieve the dual faces that intercept the node neighbors;
- Define B_j^*, the initial support region boundary, as the set of fine cells part of a dual face that contains a background grid face in the boundary of the node neighbors;
- Set $I_j \leftarrow C_j$ as the initial support region;
- **while** *there are fine-scale volumes to be incorporated to the support region* **do**
  - $I_j^+ \leftarrow$ the fine-scale volumes that share at least a face with a fine volume in I_j;
  - **if** $I_j^+ - I_j - B_j^* \neq \varnothing$ **then**
    - $I_j \leftarrow I_j \cup (I_j^+ - B_j^*)$;
  - **else**
    - Stop;
  - **end**
- **end**
- Set B_j as the support region's boundary by filtering from B_j^* all the fine cells that do not share a face with another cell outside of I_j;
- Add the cells filtered on the previous step to I_j;

end

Figure 7 – An example of preliminary support boundary (red) generated during the execution of Algorithm 6. The dark bold lines delimit the background grid neighbors of the volume of interest (lighter lines). The exceeding cells are circled in blue.



Source: Author.

6 RESULTS AND DISCUSSION

In this chapter, we present a series of examples to evaluate the performance of the proposed 3-D MsCV and E-MsCV. In the first test, we have manufactured a simple problem with a 1-D linear solution in a conical domain in order to evaluate the performance of the 3-D MsCV in a homogeneous and isotropic medium and showcase the ability of the background grid framework to generate the multiscale grids on unconventional geometries. The second example was adapted from EYMARD et al. (2011) in a homogeneous and mildly anisotropic medium. In the third example, we consider an anisotropic and heterogeneous medium with a constant source term. In the last example, we consider a reservoir with a spherical heterogeneity and discuss the multiscale solutions on a barrier and a channel configurations. For all examples, we have performed simulations using both the 3-D MsCV and the E-MsCV.

We define the following error norms for the multiscale solution (SOUZA et al., 2022):

$$||\mathbf{p}^{ref} - \mathbf{p}^{ms}||_2 = \left(\frac{\sum_{\Omega_i \in \Omega_f} |p_i^{ref} - p_i^{ms}|^2}{\sum_{\Omega_i \in \Omega_f} |p_i^{ref}|^2} \right)^{1/2}, \tag{6.1}$$

$$||\mathbf{p}^{ref} - \mathbf{p}^{ms}||_\infty = \frac{\max_{\Omega_i \in \Omega_f} |p_i^{ref} - p_i^{ms}|}{\max_{\Omega_i \in \Omega_f} |p_i^{ref}|}, \tag{6.2}$$

where the superscripts ms and ref correspond to the multiscale solution and the reference fine-scale solution, respectively. No smoothing procedures to improve the multiscale solutions were adopted.

For the timing results, all tests were run in a system with Intel® Core™ i5-8250U CPU and 8 GB RAM memory. To solve the fine-scale system of equations, the SciPy's (VIRTANEN et al., 2020) implementation of the Generalized Minimal Residual Method (GMRES) was used. Furthermore, the time considered is the average of 10 executions of the same problem in the same grid and, for the E-MsCV executions, a structured hexahedral coarse grid is used to generate the MsCV entities such that the coarsening ratio is fixed at each experiment.

6.1 Homogeneous and isotropic medium in a cone-shaped domain

In the first example, we study the simulation of a single-phase incompressible flow in a homogeneous and isotropic medium. The domain has the shape of a cone around the

Cartesian z-axis with height and radius equal to 1 and 3, respectively. We consider the following exact solution:

$$u(x, y, z) = x, \tag{6.3}$$

with permeability tensor given by:

$$\underset{\sim}{\mathcal{K}}(x, y, z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.4}$$

For all problems solved here, Dirichlet boundary conditions are applied to the whole domain boundary.

For this simulation, we have used a fine-scale grid containing 119,840 cells and an unstructured background grid with 272 tetrahedra. In Figure 8, the multiscale coarse grids used for the simulation are shown. The background grid based pre-processing framework is capable of generating coherent unstructured grids suited for simulation even in unconventionally shaped domains, as it is the case.

Figure 8 – The multiscale coarse grids used for the simulation in the cone-shaped domain.

(a) Primal coarse grid.

(b) Dual coarse grid (red).

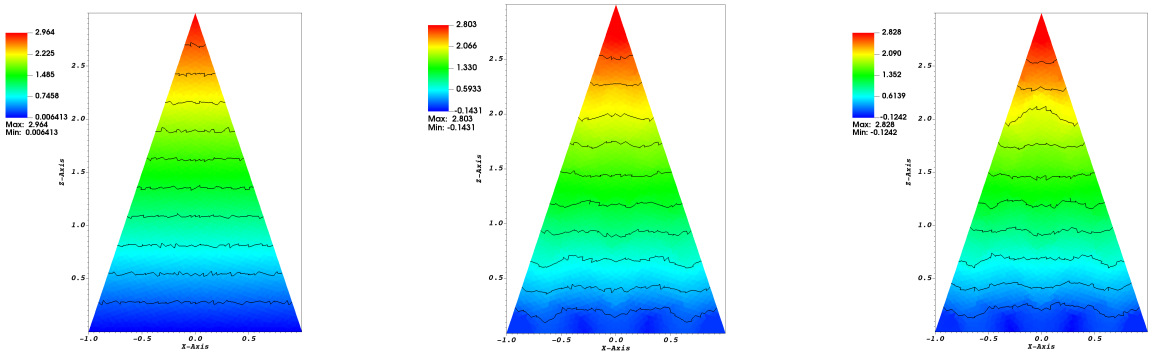
Source: Author.

Table 1 presents the L_2 and L_∞ norms of the error calculated for the MsCV and E-MsCV executions. Both approaches have similar performances, with the L_2 norm of the error lower than 0.5% while the L_∞ norm of the error stays around 7%. Figure 9 also shows that both solutions are qualitatively close to each other and to the reference solution. However, the solution obtained using the E-MsCV converges much faster than that found using the original MsCV technique. While the latter takes almost 500 iterations to converge, the former reaches approximately the same error rate with just over 100 iterations.

Table 1 – The L_2 and L_∞ norms of the errors for the homogeneous and isotropic medium in a cone-shaped domain case.

Error (%)	MsCV	E-MsCV
$\|\|u\|\|_2$	0.11	0.14
$\|\|u\|\|_\infty$	7.73	7.00

Figure 9 – The fine-scale reference solution (a) and the multiscale solutions using the MsCV (b) and the E-MsCV (c) techniques. Slice at $y = 0$ highlighting the contour lines.



(a) Fine-scale solution. (b) MsCV solution. (c) E-MsCV solution.

Source: Author.

6.2 Homogeneous and mildly anisotropic case

For this example, we consider the Test Case 1 proposed by EYMARD et al. (2011) with a regular solution over the domain $\Omega = [0, 1]^3$ implying in a non-homogeneous Dirichlet condition over the whole domain boundary Γ:

$$u(x, y, z) = 1 + \sin(\pi x) \sin\left(\pi\left(y + \frac{1}{2}\right)\right) \sin\left(\pi\left(z + \frac{1}{3}\right)\right), \tag{6.5}$$

and an anisotropic permeability tensor:

$$\underset{\sim}{\mathcal{K}}(x, y, z) = \begin{pmatrix} 1 & 0.5 & 0 \\ 0.5 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{pmatrix}. \tag{6.6}$$

We have executed the simulations on a fine-scale grid containing 243,840 tetrahedral cells and, to compute the multiscale solutions, we have used a $6 \times 6 \times 6$ structured hexahedral background grid, which corresponds to a coarsening ratio of around 1,129, and an unstructured tetrahedral background grid with 192 cells, coarsening ratio of around 1,270. For each grid, we performed simulations with the MsCV and E-MsCV.

In Table 2, the L_2 and L_∞ norms of the error from each execution are presented. In all scenarios, the L_2 norm of the error is inferior to 0.5% with a reasonable but significant error on the L_∞ norm. Nevertheless, as it can be seen in Figure 10, the solutions are qualitatively close to the reference solution in Figure 10a and are satisfactory to good approximations of the fine-scale solution. In this example, the multiscale solutions on a structured background grid produce lower errors than their unstructured counterparts. This is partially justified by the fact that, in a homogeneous medium, the support regions generated in a structured background grid are more compact and produce basis functions that can transition more smoothly between them. In an unstructured background grid, on the other hand, the coarse cells may be unaligned to the solution and the permeability tensor, affecting the shape of the basis functions and, by extension, the final solution.

With respect to the performances of the proposed methods, it is notable that both the original MsCV strategy and the E-MsCV version have similar error rates. However, from our experiments, the E-MsCV has a better convergence rate, taking 121 iterations while the original MsCV technique needed to be halted after 250 iterations. In addition, the MsCV is subject to oscillations during the iterative process, which was not observed in the E-MsCV.

Table 2 – The L_2 and L_∞ norms of the errors for the homogeneous and mildly anisotropic case.

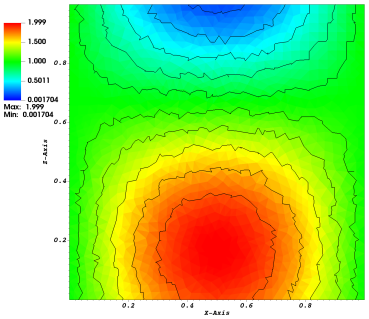
Error (%)	**Structured background grid**		**Unstructured background grid**	
	MsCV	E-MsCV	MsCV	E-MsCV
$\|\|u\|\|_2$	0.29	0.18	0.40	0.28
$\|\|u\|\|_\infty$	11.81	8.69	16.12	14.97

For this example, we also analyse the computational cost to simulate the problem by solving the original system of equations and by using the E-MsCV. The timing results presented in Figure 11 show that the E-MsCV tends to perform similarly to the fine-scale solution, except for one of the runs in which the E-MsCV outperforms the GMRES solver by a factor of 3. Since the problem is homogeneous and only mildly anisotropic, the classical GMRES can find a solution within a reasonable time, not justifying the use of an approximate multiscale solution.

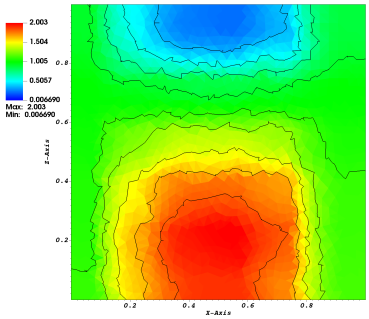
6.3 Heterogeneous and anisotropic case

In this example, we consider an adaptation of the third example from SOUZA et al. (2022) to study the robustness of the MsCV in a heterogeneous and anisotropic medium. The problem consists in the simulation of a single-phase flow in a unitary cubic domain $\Omega = [0, 1]^3$ with the following source term:

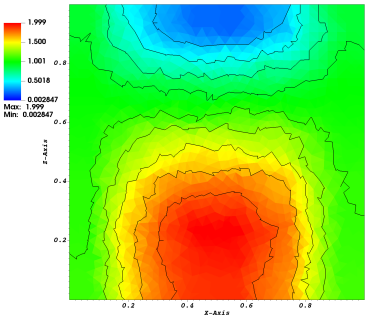
Figure 10 – The fine-scale (a) and multiscale (b-e) solutions of the homogeneous and mildly anisotropic case highlighting the contour curves. Slice at $y = 0$.



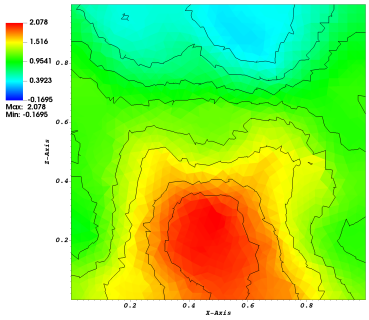
(a) Fine-scale reference solution.



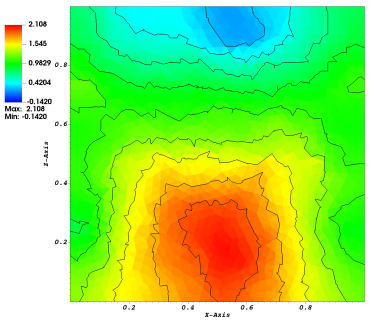
(b) Structured background grid, MsCV solution.



(c) Structured background grid, E-MsCV solution.



(d) Unstructured background grid, MsCV solution.



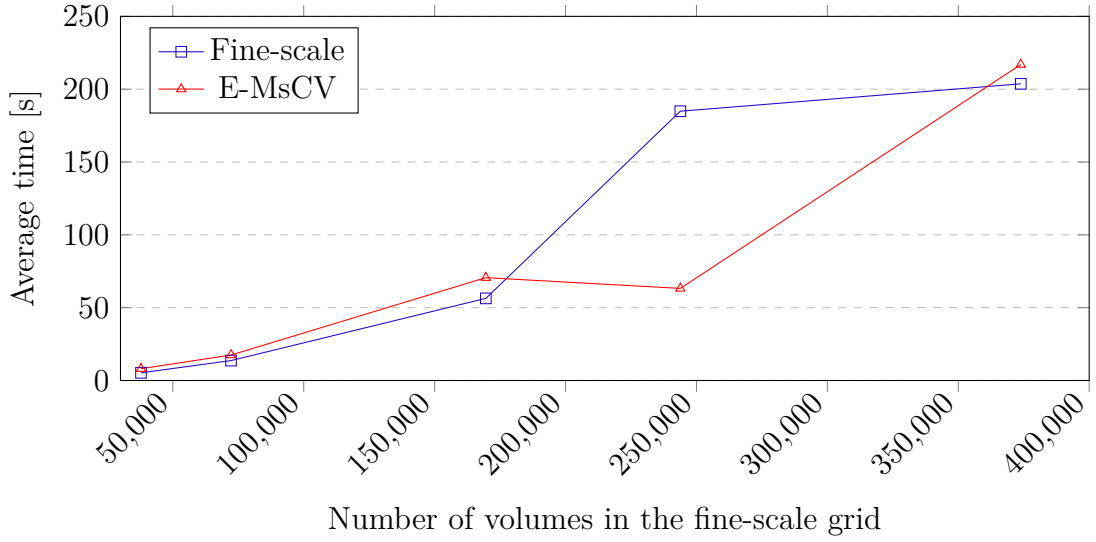
(e) Unstructured background grid, E-MsCV solution.

Source: Author.

$$Q(\overrightarrow{x}) = \begin{cases} 1, & \overrightarrow{x} \in \left[\frac{3}{8}, \frac{5}{8}\right]^3, \\ 0, & \text{otherwise} \end{cases} . \tag{6.7}$$

The region where the source term is set can be seen in Figure 12. At the boundary, the Dirichlet condition $g_D = 0$ is adopted. Furthermore, the permeability tensor is the same proposed by EYMARD et al. (2011) in the Benchmark Test Case 2, as it can be

Figure 11 – Average time to obtain a solution to the Homogeneous and mildly anisotropic case by solving the fine-scale system of equations (blue) and via the E-MsCV method (red).



Source: Author.

seen in Figure 13, and is given by:

$$\underset{\sim}{\mathcal{K}}(x,y,z) = \begin{pmatrix} y^2+z^2+1 & -xy & -xz \\ -yx & x^2+z^2+1 & -yz \\ -zx & -zy & x^2+y^2+1 \end{pmatrix}. \tag{6.8}$$

The simulations were conducted in a fine-scale grid with 90,831 tetrahedral cells.

In Table 3, the L_2 and L_∞ norms of the error are displayed. The 3-D MsCV and the E-MsCV both perform well in the L_2 norm, with a relative error inferior to 6%. Figure 14 shows that the multiscale solutions are qualitatively close to the fine-scale reference solution, specially on a structured $6 \times 6 \times 6$ hexahedral background grid.

From Table 3, it is noticeable that both methods produce high errors on the L_∞ norm. Although this problem does not have an analytical solution, by the discrete maximum principle (DMP), the approximate solution should be greater than 0 in all domain. In Figure 14, we can see that the multiscale is prone to spurious oscillations, depending on the background grid. It also produces negative pressure values on all cases. However, the fine-scale MPFA-D solution itself violates the DMP, and the oscillatory behaviour seems to be propagated and amplified by the multiscale basis functions.

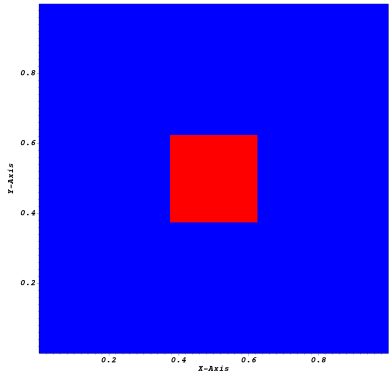
Still regarding Figure 14, it is possible to see that the solutions on the $6 \times 6 \times 6$ background grid is able to better capture the diffusion pattern of the reference solution. Besides the fact that it is a more refined grid, from Figure 15, we can see that the dual grid generated from the $5 \times 5 \times 5$ background grid contains faces crossing the region where the source term is non null. This affects the quality of the basis functions since the source

term is spread among multiple volumes instead of being captured mostly within a single support region.

Table 3 – The L_2 and L_∞ norms of the errors for the heterogeneous and anisotropic case.

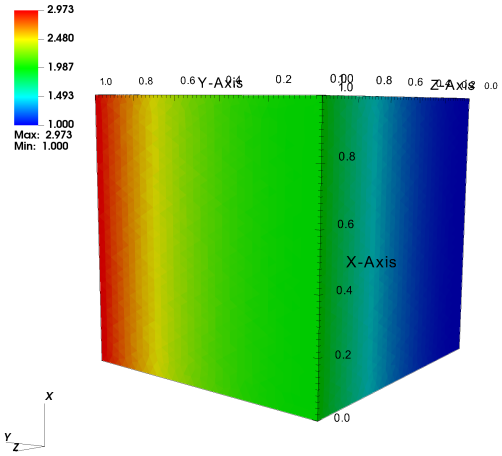
Error (%)	$5 \times 5 \times 5$ **background grid**		$6 \times 6 \times 6$ **background grid**	
	MsCV	E-MsCV	MsCV	E-MsCV
$\|\|u\|\|_2$	5.04	3.95	1.93	1.84
$\|\|u\|\|_\infty$	30.18	53.68	16.04	16.93

Figure 12 – The region within the reservoir where the source term is set (red). Slice at $z = 0.5$.



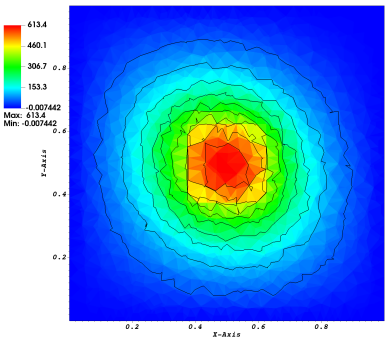
Source: Author.

Figure 13 – Visualization of the permeability field (K_{xx}) of the heterogeneous and anisotropic case.

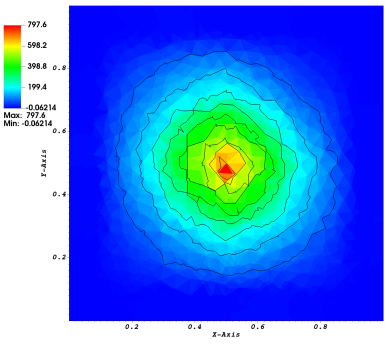


Source: Author.

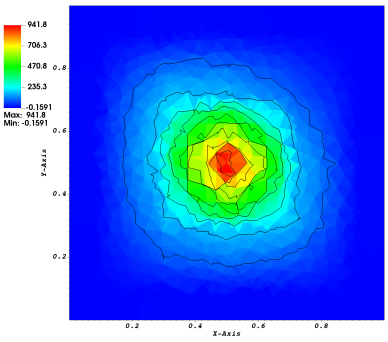
Figure 14 – The fine-scale (a) and multiscale (b-e) solutions of the heterogeneous and anisotropic case highlighting the contour curves. Slice at $z = 0.5$.



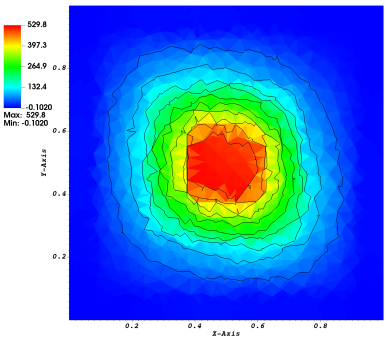
(a) Fine-scale reference solution.



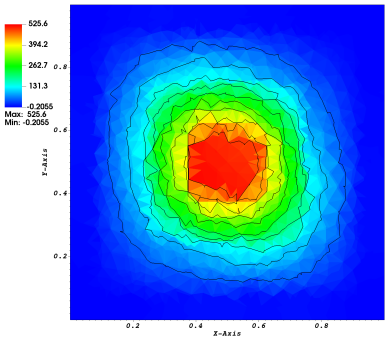
(b) Structured 5×5×5 background grid, MsCV solution.



(c) Structured $5 \times 5 \times 5$ background grid, E-MsCV solution.



(d) Structured 6×6×6 background grid, MsCV solution.



(e) Structured $6 \times 6 \times 6$ background grid, E-MsCV solution.

Source: Author.

6.4 Reservoir with a spherical heterogeneity

In this final example, we study the simulation of a single-phase flow in a reservoir containing a spherical heterogeneity within the domain $\Omega = [-2, 2]^3$. The following boundary conditions are applied:

Figure 15 – The dual coarse grids generated for the heterogeneous and anisotropic case highlighting the background grid (black lines) and the region where the source term is non null (blue line). Slices at $z = 0.6$ for (a) and $z = 0.5$ for (b).

(a) Structured $5 \times 5 \times 5$ background grid.

(b) Structured $6 \times 6 \times 6$ background grid.

Source: Author.

$$\begin{cases} g_D = 0 & \text{on } \Gamma_{D,1} \\ g_D = 1 & \text{on } \Gamma_{D,2} \\ g_N = 0 & \text{on } \Gamma_N \end{cases}, \tag{6.9}$$

where $\Gamma_{D,1}$ corresponds to the planes $x = -2$, $\Gamma_{D,2}$ corresponds to the plane $x = 2$, and Γ_N is set at the planes $y = -2$, $y = 2$, $z = -2$ and $z = 2$.

The permeability field is illustrated in Figure 16, where the heterogeneity region shaped as a sphere centered at the origin with a radius equal to 0.75 is embedded in a homogeneous domain with a permeability tensor given by:

$$\mathcal{K}_1(x, y, z) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{6.10}$$

The goal of this example is to test the changes in the solution when applying background grids that are conforming or not to the heterogeneity, and evaluate the ability of the 3-D MsCV to capture such formation. We simulate two configurations, a barrier and a channel, with permeability tensors respectively given by:

$$\underset{\sim}{\mathcal{K}}_2(x,y,z) = \begin{pmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 10^{-3} \end{pmatrix}, \tag{6.11}$$

$$\underset{\sim}{\mathcal{K}}_3(x,y,z) = \begin{pmatrix} 10^{3} & 0 & 0 \\ 0 & 10^{3} & 0 \\ 0 & 0 & 10^{3} \end{pmatrix}. \tag{6.12}$$

The simulations were conducted in a fine-scale grid with 159,893 tetrahedral cells.

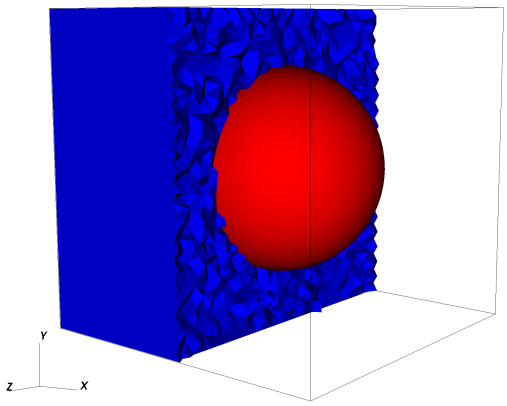
The multiscale solution for the channel configuration is presented in Figure 18 for different background grid topologies. In all cases, the solutions remain qualitatively close to the fine-scale reference solution shown in Figure 17b. Although the results obtained with the unstructured grid are able to better capture the channel's geometry, it presents a higher error on the boundaries. The errors in Table 4 corroborate with the previous observations, as the L_2 norm of the error is approximately 0.6% with the structured background grid and below 4% with the unstructured one despite the high values for the L_∞ norm of errors. On both grids, the performance of the MsCV and E-MsCV are still comparable. As with the previous examples, the original MsCV's iterations present a slower convergence rate and did not converge on the prescribed tolerance criterion equal to 10^{-3}, requiring to stop the iterative procedure after 500 iterations to achieve a result with errors of the same magnitude as those presented by the E-MsCV, which in turn took 132 iterations.

In Figure 19, the multiscale solution for the barrier configuration is displayed. In this configuration, the solution using the original MsCV preconditioning fails to converge regardless of the background grid used. On the other hand, the E-MsCV converges. Albeit not able to fully capture the barrier in the reservoir, the E-MsCV's solution still manages to reasonably show its main features on both background grids. As seen in Table 5, the L_2 norm of the errors on the solution using the E-MsCV are still satisfactory even though the L_∞ norm of the error is very high.

Table 4 – The L_2 and L_∞ norms of the errors for the simulation of a reservoir containing a spherical heterogeneity under a channel configuration.

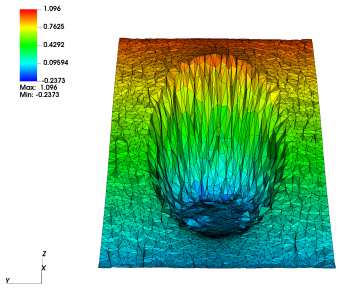
Error (%)	**Structured background grid**		**Unstructured background grid**	
	MsCV	E-MsCV	MsCV	E-MsCV
$\|\|u\|\|_2$	0.69	0.59	2.00	3.62
$\|\|u\|\|_\infty$	18.79	19.04	33.87	49.30

Figure 16 – The spherical heterogeneity region (red) within the reservoir.

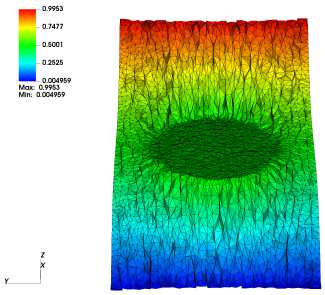


Source: Author.

Figure 17 – Fine-scale reference solutions for the single-phase simulation of a reservoir containing a spherical heterogeneity under a barrier (a) and channel (b) configuration. Slice at $y = 0$.



(a) Barrier configuration.



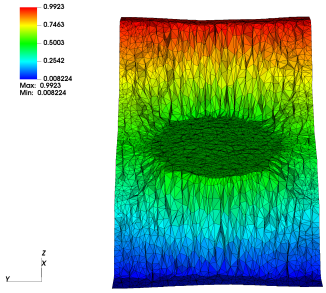
(b) Channel configuration.

Source: Author.

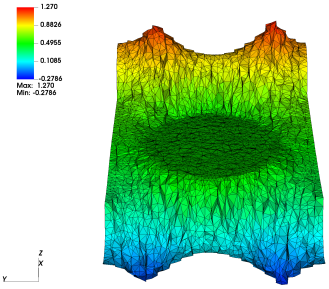
Table 5 – The L_2 and L_∞ norms of the errors for the simulation of a reservoir containing a spherical heterogeneity under a barrier configuration.

Error (%)	**Structured background grid**		**Unstructured background grid**	
	MsCV	E-MsCV	MsCV	E-MsCV
$\|\|u\|\|_2$	7×10^8	6.02	7×10^5	6.93
$\|\|u\|\|_\infty$	5×10^7	83.78	8×10^6	86.79

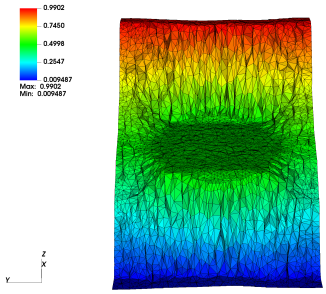
Figure 18 – Multiscale solutions under a channel configuration using different background grids and preconditioning techniques. Slice at $y = 0$.



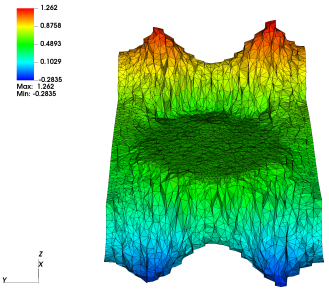
(a) Structured background grid, MsCV solution.



(b) Unstructured background grid, MsCV solution.



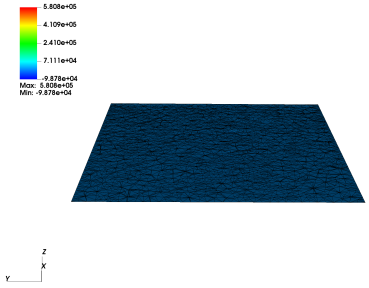
(c) Structured background grid, E-MsCV solution.



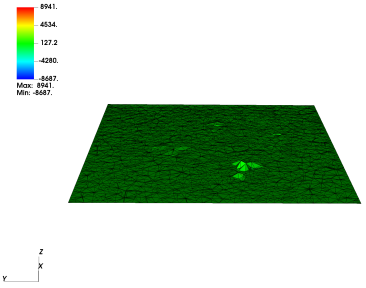
(d) Unstructured background grid, E-MsCV solution.

Source: Author.

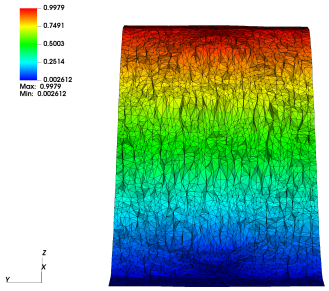
Figure 19 – Multiscale solutions under a barrier configuration using different background grids and preconditioning techniques. Slice at $y = 0$.



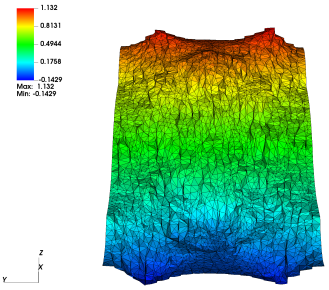
(a) Structured background grid, MsCV solution.



(b) Unstructured background grid, MsCV solution.



(c) Structured background grid, E-MsCV solution.



(d) Unstructured background grid, E-MsCV solution.

Source: Author.

7 CONCLUSIONS

In this thesis, we have presented a 3-D extension to the Multiscale Control Volume (MsCV) method coupling it with the 3-D MPFA-D and the robust GLS interpolation for the simulation of single-phase flow in heterogeneous and anisotropic porous media. In addition, we have proposed an extension to 3-D geometries of the very flexible multiscale pre-processing algorithm by SOUZA et al. (2022) based on the concept of a background grid and an enhanced version of the 3-D MsCV (E-MsCV) that incorporates the enhanced MsRSB preconditioning (BOSMA et al., 2021) to improve the convergence.

From our experiments, the 3-D MsCV is capable of approximating the reference fine-scale solution to a good degree on low to intermediate complexity scenarios and the E-MsCV is able to converge on more challenging scenarios. It was also possible to notice that, despite of the robustness of the MPFA-D with the GLS interpolation, there are violations of the DMP which are exacerbated in the multiscale solution. Finally, the background grid framework was able to correctly generate the multiscale grids in the tested scenarios, working with truly unstructured fine-scale and background grids. We have noticed that the background grid framework may show some limitations when faced with a fine-scale grid that contains highly deformed elements, generating inconsistent dual coarse grids.

Herein are some suggestions for future works:

- Study smoothing techniques to improve the 3-D MsCV's solution;
- Extend the framework to the simulation of multiphase flow in porous media;
- Explore new approaches to generate the multiscale geometric entities under the background grid framework by adapting it to the underlying geological characteristics;
- Fix the inconsistencies observed in the background grid-based pre-processing when applied to very distorted grids;
- Investigate the DMP violations on the multiscale solution by introducing a defect correction scheme as proposed by (CAVALCANTE et al., 2022; SOUZA et al., 2023);
- Probe algebraic multiscale strategies in place of the MsRSB;
- Examine the use of the multiscale operators and solution as a pre-conditioner to the fine-scale system of equations;
- Study HPC techniques to enable the multiscale framework for large-scale simulations.

REFERENCES

AAVATSMARK, I. et al. Discretization on unstructured grids for inhomogeneous, anisotropic media. part i: Derivation of the methods. *SIAM Journal on Scientific Computing*, v. 19, n. 5, p. 1700–1716, 1998. Disponível em: <https://doi.org/10.1137/S1064827595293582>.

AAVATSMARK, I. et al. Discretization on unstructured grids for inhomogeneous, anisotropic media. part ii: Discussion and numerical results. *SIAM Journal on Scientific Computing*, v. 19, n. 5, p. 1717–1736, 1998. Disponível em: <https://doi.org/10.1137/S1064827595293594>.

ARBOGAST, T.; BRYANT, S. L. A Two-Scale Numerical Subgrid Technique for Waterflood Simulations. *SPE Journal*, v. 7, n. 04, p. 446–457, 12 2002. ISSN 1086-055X. Disponível em: <https://doi.org/10.2118/81909-PA>.

BARBOSA, L. M. C. et al. An iterative modified multiscale control volume method for the simulation of highly heterogeneous porous media flow. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 40, n. 71, 2018. Disponível em: <https://doi.org/10.1007/s40430-017-0929-z>.

BORSUK, M.; KONDRATIEV, V. Chapter 5 - the dirichlet problem for elliptic linear divergent equations in a nonsmooth domain. In: BORSUK, M.; KONDRATIEV, V. (Ed.). *Elliptic Boundary Value Problems of Second Order in Piecewise Smooth Domains*. Elsevier, 2006, (North-Holland Mathematical Library, v. 69). p. 165–213. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0924650906800188>.

BOSMA, S. et al. Multiscale finite volume method for discrete fracture modeling on unstructured grids (ms-dfm). *Journal of Computational Physics*, v. 351, p. 145–164, 2017. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999117306927>.

BOSMA, S. B. et al. Enhanced multiscale restriction-smoothed basis (msrsb) preconditioning with applications to porous media flow and geomechanics. *Journal of Computational Physics*, v. 428, p. 109934, 2021. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999120307087>.

CARVALHO, D. K. E. de. *Uma Formulação do Método dos Volumes Finitos com Estrutura de Dados por Aresta para a Simulação de Escoamentos em Meios Porosos*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2005.

CAVALCANTE, T. de M. *Simulation of Immiscible Two-Phase Flow in 3-D Naturally Fractured Reservoirs Using a Locally Conservative Method, a Projection-Based Embedded Discrete Fracture Model and Unstructured Tetrahedral Meshes*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2023.

CAVALCANTE, T. M. et al. A multipoint flux approximation with a diamond stencil and a non-linear defect correction strategy for the numerical solution of steady state diffusion problems in heterogeneous and anisotropic media satisfying the discrete maximum

principle. *Journal of Scientific Computing*, v. 93, n. 2, p. 42, Sep 2022. ISSN 1573-7691. Disponível em: <https://doi.org/10.1007/s10915-022-01978-6>.

CHEN, Z.; HOU, T. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Mathematics of Computation*, v. 72, n. 242, p. 541–576, 2003.

CONTRERAS, F. et al. A cell-centered multipoint flux approximation method with a diamond stencil coupled with a higher order finite volume method for the simulation of oil–water displacements in heterogeneous and anisotropic petroleum reservoirs. *Computers & Fluids*, v. 127, p. 1–16, 2016. ISSN 0045-7930. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045793015003849>.

CORTINOVIS, D. *Robust multiscale finite volume method based iterative schemes*. Tese (Doutorado) — ETH Zurich, 2016.

DELAUNAY, B. et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, v. 7, n. 793-800, p. 1–2, 1934.

DONG, C.; KANG, T. A least squares based diamond scheme for 3d heterogeneous and anisotropic diffusion problems on polyhedral meshes. *Applied Mathematics and Computation*, v. 418, p. 126847, 2022. ISSN 0096-3003. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0096300321009309>.

DURLOFSKY, L. J. Upscaling and gridding of fine scale geological models for flow simulation. In: *8th International forum on reservoir simulation*. Iles Borromees, Stresa, Italy: [s.n.], 2005. v. 2024.

EDWARDS, M. G.; ZHENG, H. A quasi-positive family of continuous darcy-flux finite-volume schemes with full pressure support. *Journal of Computational Physics*, v. 227, n. 22, p. 9333–9364, 2008. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999108002908>.

EWING, R. E. *The Mathematics of Reservoir Simulation*. Society for Industrial and Applied Mathematics, 1983. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9781611971071>.

EYMARD, R. et al. 3d benchmark on discretization schemes for anisotropic diffusion problems on general grids. In: FOŘT, J. et al. (Ed.). *Finite Volumes for Complex Applications VI Problems & Perspectives*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 895–930. ISBN 978-3-642-20671-9.

FARMER, C. Upscaling: a review. *International journal for numerical methods in fluids*, Wiley Online Library, v. 40, n. 1-2, p. 63–78, 2002.

GAO, Z.; WU, J. A linearity-preserving cell-centered scheme for the heterogeneous and anisotropic diffusion equations on general meshes. *International Journal for Numerical Methods in Fluids*, v. 67, n. 12, p. 2157–2183, 2011. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2496>.

HAGBERG, A. A.; SCHULT, D. A.; SWART, P. J. Exploring network structure, dynamics, and function using networkx. In: VAROQUAUX, G.; VAUGHT, T.; MILLMAN, J. (Ed.). *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA: [s.n.], 2008. p. 11 – 15.

HAJIBEYGI, H. et al. Iterative multiscale finite-volume method. *Journal of Computational Physics*, v. 227, n. 19, p. 8604–8621, 2008. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S002199910800332X>.

HARRIS, C. R. et al. Array programming with NumPy. *Nature*, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>.

HOU, T. Y.; WU, X.-H. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, v. 134, n. 1, p. 169–189, 1997. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999197956825>.

JARAMILLO, A. et al. Towards hpc simulations of billion-cell reservoirs by multiscale mixed methods. *Computational Geosciences*, v. 26, n. 3, p. 481–501, Jun 2022. ISSN 1573-1499. Disponível em: <https://doi.org/10.1007/s10596-022-10131-z>.

JENNY, P.; LEE, S.; TCHELEPI, H. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, v. 187, n. 1, p. 47–67, 2003. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999103000755>.

KARYPIS, G.; KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, v. 20, n. 1, p. 359–392, 1998. Disponível em: <https://doi.org/10.1137/S1064827595287997>.

LEE, S. H.; WOLFSTEINER, C.; TCHELEPI, H. A. Multiscale finite-volume formulation for multiphase flow in porous media: black oil formulation of compressible, three-phase flow with gravity. *Computational Geosciences*, v. 12, n. 3, p. 351–366, Sep 2008. ISSN 1573-1499. Disponível em: <https://doi.org/10.1007/s10596-007-9069-3>.

LIRA FILHO, R. J. de et al. A linearity-preserving finite volume scheme with a diamond stencil for the simulation of anisotropic and highly heterogeneous diffusion problems using tetrahedral meshes. *Computers & Structures*, v. 250, p. 106510, 2021. ISSN 0045-7949. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045794921000328>.

LUNATI, I.; JENNY, P. Multiscale finite-volume method for compressible multiphase flow in porous media. *Journal of Computational Physics*, v. 216, n. 2, p. 616–636, 2006. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999106000039>.

MAHADEVAN, V. et al. *MOAB v5.2.1*. Zenodo, 2020. Disponível em: <https://doi.org/10.5281/zenodo.2584862>.

MEHRDOOST, Z. Unstructured grid adaptation for multiscale finite volume method. *Computational Geosciences*, v. 23, n. 6, p. 1293–1316, Dec 2019. ISSN 1573-1499. Disponível em: <https://doi.org/10.1007/s10596-019-09878-9>.

MØYNER, O.; LIE, K.-A. A multiscale restriction-smoothed basis method for high contrast porous media represented on unstructured grids. *Journal of Computational Physics*, v. 304, p. 46–71, 2016. ISSN 0021-9991. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0021999115006725>.

QUEIROZ, L. et al. On the accuracy of a nonlinear finite volume method for the solution of diffusion problems using different interpolations strategies. *International Journal for Numerical Methods in Fluids*, v. 74, n. 4, p. 270–291, 2014. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3850>.

ROSSUM, G. van; BOER, J. de. Interactively testing remote servers using the python programming language. *CWI Quarterly*, v. 4, n. 4, p. 283–304, dez. 1991.

SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2.. ed. Society for Industrial and Applied Mathematics, 2003. Disponível em: <https://epubs.siam.org/doi/abs/10.1137/1.9780898718003>.

SOUZA, A. C. R. et al. A non-linear repair technique based on a flux limited splitting. No prelo. 2023.

SOUZA, A. C. R. d. *IMPRESS: Intuitive Multilevel Preprocessor for Smart Simulation*. 2020. Https://github.com/padmec-reservoir/impress. Acesso em 10 jul. 2023.

SOUZA, A. C. R. de et al. A multiscale control volume framework using the multiscale restriction smooth basis and a non-orthodox multi-point flux approximation for the simulation of two-phase flows on truly unstructured grids. *Journal of Petroleum Science and Engineering*, v. 188, p. 106851, 2020. ISSN 0920-4105. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0920410519312677>.

SOUZA, A. C. R. de et al. An algebraic multiscale solver for the simulation of two-phase flow in heterogeneous and anisotropic porous media using general unstructured grids (ams-u). *Applied Mathematical Modelling*, v. 103, p. 792–823, 2022. ISSN 0307-904X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0307904X21005552>.

TAUTGES, T. J. et al. *MOAB: A Mesh-Oriented Database*. [S.l.], 2004. Report.

VÉRON, L. Chapter 8 - elliptic equations involving measures. In: CHIPOT, M.; QUITTNER, P. (Ed.). North-Holland, 2004, (Handbook of Differential Equations: Stationary Partial Differential Equations, v. 1). p. 593–712. Disponível em: <https://www.sciencedirect.com/science/article/pii/S187457330480010X>.

VIRTANEN, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, v. 17, p. 261–272, 2020.

ZHOU, H. *Algebraic multiscale finite-volume methods for reservoir simulation*. Tese (Doutorado) — Stanford University, 2010.

ZHOU, H.; TCHELEPI, H. A. Operator-Based Multiscale Method for Compressible Flow. *SPE Journal*, v. 13, n. 02, p. 267–273, 06 2008. ISSN 1086-055X. Disponível em: <https://doi.org/10.2118/106254-PA>.

Two-Stage Algebraic Multiscale Linear Solver for Highly Heterogeneous Reservoir Models, All Days de *SPE Reservoir Simulation Conference*, (SPE Reservoir Simulation Conference, All Days). Disponível em: <https://doi.org/10.2118/141473-MS>.

APPENDIX A – IMPLEMENTATION ISSUES

In this appendix, the architecture and implementation of the proposed methods is presented, as the libraries and packages that were employed to enable the development process.

A.1 Libraries and packages

The proposed methods were developed in Python language (ROSSUM; BOER, 1991) with the help of some libraries and packages. Python was the language of choice for its high-level syntax, allowing for a quick development without having to worry about low-level operations, and the wide availability of libraries for scientific computing with active community support. In this section, we will present the packages used during the code development.

A.1.1 NumPy

NumPy (HARRIS et al., 2020) is an open-source Python library that implements algorithms and data structures to handle and operate with multidimensional arrays and matrices. It provides an efficient way to perform numerical operations with large collections of values. In addition, it is widely used in the context of scientific computing and it has a large active community of users.

For the proposed implementation, NumPy's data structures were used as the basis for representing vectors and matrices, and efficiently computing operations on large sets of values.

A.1.2 SciPy

SciPy (VIRTANEN et al., 2020) is a collection of mathematical algorithms and convenience functions implemented using NumPy. It provides a wide range of functionalities from sparse matrices algorithms, to graph representation and linear systems solvers. In this thesis, we primarily used the modules for sparse matrices and the solvers for linear systems of equations.

A.1.3 NetworkX

NetworkX (HAGBERG; SCHULT; SWART, 2008) is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Here, the graph algorithms and data structures were used in the multiscale pre-processing algorithm, namely the algorithms to check the connected components of a graph.

A.1.4 IMPRESS

IMPRESS (Intuitive Multilevel Preprocessor for Smart Simulation) (SOUZA, 2020) is an open source Python package for mesh management and pre-processing developed within the PADMEC/LCCV research group. The IMPRESS is based on the PyMOAB package, a Python binding of the C/C++ library MOAB (Mesh Oriented Database) (TAUTGES et al., 2004; MAHADEVAN et al., 2020) which implements data structures and routines to represent and manipulate unstructured grids.

In the present work, IMPRESS was used for the management of the grids involved in the implementation. The data structures and functions provided allow to efficiently retrieve useful information such as adjacencies, neighbors, areas and volumes. Moreover, IMPRESS allows to easily associate data to elements in the mesh and later recover it.

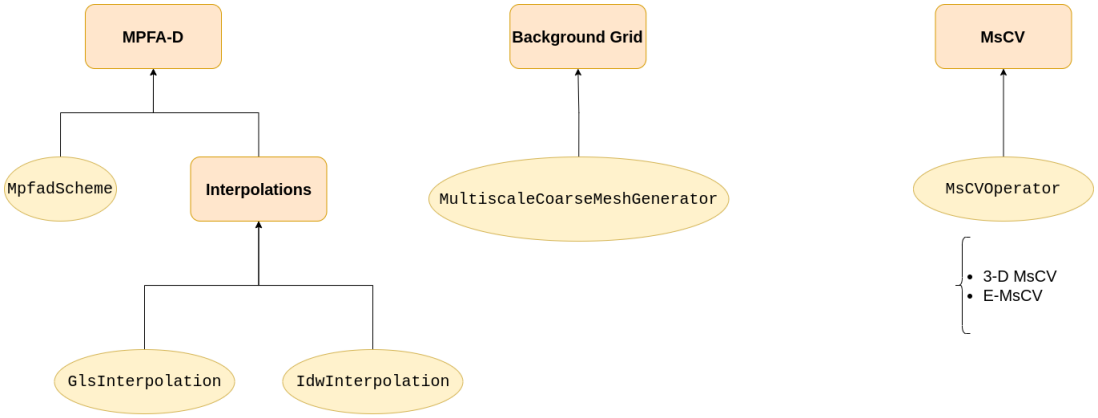
A.2 Architecture of the computational implementation

In this thesis, three methods were implemented: the 3-D MPFA-D, the background grid pre-processing framework, and the 3-D MsCV. Therefore, following this division, the computational code was also split into three major modules, as seen in the diagram of Figure 20:

- The MPFA-D module, including the implementation of the scheme itself and the nodal interpolations;
- The background grid module, which contains all data structures and procedures related to the multiscale pre-processing algorithm; and
- The MsCV module, consisting of the implementation of the multiscale operators and the pre-conditioning techniques discussed in Chapter 4.

The communication between the modules is done via file input/output, that is, the byproducts of a module are passed to another using files. This way, they can operate

Figure 20 – Simplified architecture of the implementation of the proposed methods showing the modules (orange boxes) and the main classes (yellow ellipsis).



Source: Author.

independently from each other and it is relatively easy to replace one of the implementations as long the interface remains the same.

All modules were designed under the object-oriented programming paradigm (OOP), with each distinguishable unit implemented as a class. Furthermore, the program was designed to make use of the vectorization technique, avoiding unnecessary loops where possible and rewriting procedures as vector-matrix operations to improve performance and take advantage of NumPy.

The code for both the MPFA-D and the multiscale framework implementations is publicly available at the `mpfad` and the `background-grid` GitHub repositories, respectively.

A.2.1 The MPFA-D module

The MPFA-D module is composed by the `MpfadScheme` class, which implements the assembly of the linear system of equations, and the sub-module of nodal interpolations. In the latter, there are two techniques implemented: the GLS, as presented in Chapter 3, and the Inverse Distance Weighting (IDW) (QUEIROZ et al., 2014).

The `MpfadScheme` class contains an instance of one of the interpolation techniques as an attribute. The interpolation technique can be swapped simply by passing a different instance to the class constructor. Similarly, a new interpolation technique can be added to the sub-module and used in the `MpfadScheme` by implementing the required interfaces.

Furthermore, the `MpfadScheme` class produces as output the MPFA-D system of linear equations exported as NumPy array files. The solution of the system of equations is not computed by the class, but, for the examples presented in this thesis, we used the SciPy's implementation of the Generalized Minimal Residual method (GMRES) (SAAD, 2003) to obtain its solution.

A.2.2 The Background Grid module

The Background Grid module comprises the structures and procedures that implement the 3-D background grid multiscale pre-processing algorithm. The main class in this module is the `MultiscaleCoarseMeshGenerator` which carries out the steps described in Chapter 5. The fine-scale mesh and the background grid are represented by IMPRESS' grid objects and they are loaded from a mesh file.

The output of the `MultiscaleCoarseMeshGenerator` class are a mesh file with properties assigning each fine-scale volume to its type in the primal and dual coarse grids, and Python's dictionaries exported as binary files containing the composition of the support region and the support boundary of each primal coarse volume.

A.2.3 The MsCV module

The final module contains the implementation of the proposed MsCV method. Here, an instance of the MsCV problem is represented by an instance of the `MsCVOperator` class. As an input, the class constructor receives both the pre-processing files and the files corresponding to the MPFA-D system of equations.

In this thesis, we proposed both the 3-D extension of the original MsCV and an enhanced version, the E-MsCV. Since the main difference between each method is the preconditioning technique used for the Jacobi iterations, both are implemented within the `MsCVOperator` class. To switch between the formulations, one needs only to specify which preconditioning will be used.