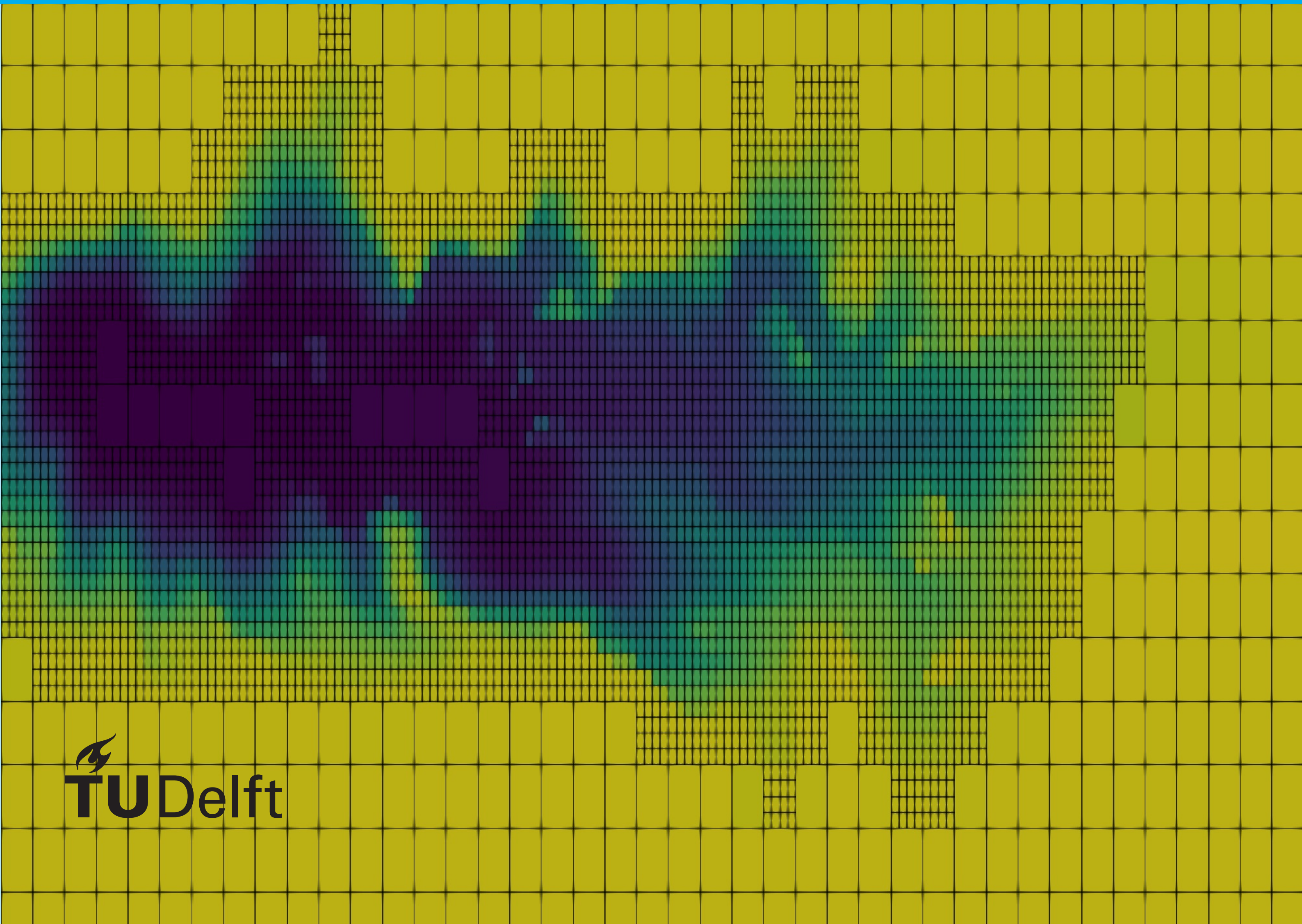
Applications of unstructured multi-level grid to thermal-reactive flow and transport in porous media

E. J. Jones



TU Delft

Applications of unstructured multi-level grid to thermal-reactive flow and transport in porous media

by

E. J. Jones

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday November 8, 2019 at 4:30 PM.

Student number:	4291573	
Project duration:	December 1, 2018 – November 8, 2019	
Thesis committee:	Dr. D. V. Voskov,	TU Delft, supervisor
	Prof. W. R. Rossen,	TU Delft
	Dr. P. J. Vardon,	TU Delft
	S. de Hoop,	TU Delft

This thesis is confidential and cannot be made public until November 8, 2019.

Abstract

A coupled description of flow and thermal-reactive transport is spanning a wide range of scales in space and time. Subsurface reservoir heterogeneity with complex multi-scale features increases the complexity further. The spatial resolution required during a simulation is dependent on the type and degree of geological heterogeneity, but also on the flow physics and the well locations (Karimi-Fard & Durlofsky, 2014). Traditional upscaling or multi-scale techniques are usually focused on the accuracy of the pressure solution and often ignore the transport. Improving the transport solution can however be quite significant for the performance of the simulation, especially in complex applications related to reactive and compositional flow. The use of a method called *Adaptive Mesh Refinement* (AMR) enables the grid to adapt dynamically during the course of the simulation, which facilitates the efficient use of computational resources (Karimi-Fard & Durlofsky, 2014; Cusini et al., 2016). In this work, the aim was to develop such an AMR framework for general-purpose reservoir simulation. The approach uses a multi-level grid and is applicable to fully unstructured grids.

The adaptivity of the grid in the developed AMR procedure is based on a hierarchical grid. The multi-level grid is constructed starting with the static geological model, a fine-scale model that accurately approximates the prospective reservoir. The control volumes present in the model are described by a list of volumes, depths and reservoir properties, and by a connectivity list for all neighbouring blocks, with corresponding transmissibility. The coarser levels are constructed through cell partitioning of the fine-scale model, called level 0. A global flow-based upscaling is applied to redefine the cell properties at the coarser levels. Each coarser level is also described by a list of volumes, depths, reservoir properties and a connectivity list with corresponding transmissibility. An inter-level connectivity list is constructed which includes the connections between control volumes of consecutive levels. The connectivity lists with corresponding transmissibility, along with the cell properties of each control volume are combined to form global arrays describing the full hierarchy of levels. The simulation is then started at the coarsest level, while keeping the wells at the finest level of refinement for more accurate representation of the solution. Dynamic adaptivity is performed, based on adaptivity criteria which were developed specific to the used application, resulting in a new grid configuration. The multi-level connectivity list together with the cell properties are redefined at each time step and used as input for the next simulation. The developed AMR framework was implemented in *Delft Advanced Research Terra Simulator* (DARTS, 2019) which uses the recently developed linearization technique called Operator-Based Linearization, developed by Voskov (2017).

The performance of the proposed approach was illustrated for several applications, including geothermal energy extraction and reactive transport. For geothermal applications, several models were tested: a homogeneous reservoir with unstructured grid, a synthetic fluvial model with shale blocks illustrating the role of shale in thermal recharge, a highly heterogeneous fluvial system with low net-to-gross ratio, and several layers of a highly-heterogeneous SPE10 reservoir. The results of these flow examples demonstrate a high level of solution accuracy relative to the reference fine-scale solution. The solution error is considerably decreased as compared to the coarse-scale model. Important features are captured at a higher resolution while keeping the amount of cells to a reasonable number. For the fluvial model, a different approach was adopted for cell aggregation, consisting of grouping the facies together, where shale-regions are kept coarse, and high-permeability facies are kept at a fine level. This significantly improved the solution as compared to the coarse model, where cold front propagation is overestimated. For the application of reactive flow, the dissolution of calcium carbonate through the formation of wormholes was analyzed. Here as well, the solution is considerably improved when using the AMR model. The wormholes are accurately represented, with improved computational performance.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor Dr. Denis Voskov for the continuous support during my Master thesis project, for his patience and time, motivation, enthusiasm and immense knowledge. He always took the time to help me when needed, regardless of how busy he was. I would also like to thank Stephan de Hoop for helping me at times with DARTS and for giving assistance and advice when I ran into problems during my project. I would like to thank committee members Prof. Bill Rossen and Dr. Phil Vardon for their time and interest in my work.

Furthermore, I would also like to acknowledge with much appreciation the SFB1313 research group from the University of Stuttgart for providing financial support for the research. I would particularly like to thank Beatrix Becker and Prof. Rainer Helmig for their valuable time and help and for giving me the opportunity to work with them in Stuttgart. I would also like to thank the whole SFB1313 research team for making my stay in Stuttgart enjoyable and for welcoming me to their team during my 2 months stay and for letting me take part in the SFB seminar and other activities.

Last but not least, I am very thankful to my family, my boyfriend and my friends for their encouragement and support throughout my studies.

Contents

List of figures

List of tables

Nomenclature

List of abbreviations

2D	Two-dimensional
3D	Three-dimensional
AMR	Adaptive Mesh Refinement
DARTS	Delft Advanced Research Terra Simulator
EOR	Enhanced Oil Recovery
FVM	Finite Volume Method
OBL	Operator Based Linearization
TPFA	Two-Point Flux Approximation

List of symbols

$\tilde{q}_p$	source of phase p
$\boldsymbol{\omega}$	function of physical state
$\boldsymbol{\xi}$	function of spatial coordinate
$\cap$	intersection of two sets
$\Delta\psi^l$	pressure difference between blocks connected via interface l
Δt	time step
ΔT^l	temperature difference between blocks connected via interface l
$\forall$	for all element
Γ^l	constant geometrical part of the transmissibility
Γ^l_c	thermal (conductive) transmissibility
Γ^l_p	phase transmissibility
$\in$	element in a given set
κ	thermal conduction
λ	mobility of a given phase p
K	permeability tensor
U_p	phase internal energy
U_r	rock internal energy
μ_p	phase viscosity
$\overrightarrow{\gamma}_p$	gravity vector
$\overrightarrow{u_p}$	phase velocity
ϕ	porosity
ρ_p	phase molar density
$\mathbf{f}$	set of faces of a given element

$\mathbf{u}$	well control variables
$\mathbf{x}$	set of vertices of a given element
X_o	solution variable at initial conditions
X_P	prolongated solution X
X_R	restricted solution X
X	solution variable (unknown we want to solve for)
A	area
D	depth (backward oriented)
D_i	distance between centroid of cell i to interface area A
d_i	unit vector along the line joining centroid of cell i to the center of interface A
h_p	phase enthalpy
J	Jacobian in the Newton-Raphson method
k	permeability
k_{rp}	relative permeability of phase p
n	current time step
n+1	next time step
n_c	number of components c
n_i	unit vector
n_k	number of kinetic reactions k
n_p	number of phases p
P	pressure
P_p	pressure in phase p
q_p	source of phase p
Q_{ij}	flow rate between cell i and j
r	residual in the Newton-Raphson method
r_k	rate of kinetic reaction k
s_p	phase saturation
T	temperature
t	time
t_D	dimensionless time (current time divided by final time)
V	control volume
v_{ck}	stoichiometric coefficient associated with kinetic reaction k
x_{cp}	mole fraction of component c in phase p
z_c	overall composition of component c

1

Introduction

Production development of prospective reservoirs includes the use of various technologies that provide information at many different scales. These scales range from core plugs being a few centimeters in size to well logs detecting properties a few meters around the well, and to seismic imaging covering a significant volume with limited resolution (few meters vertically and 10's of meters horizontally). However, time and capital limitations result in sparse direct sampling of reservoir rock and fluid properties. This is why the construction of reservoir models, through integration of these data using geostatistical reservoir description algorithms, has become a crucial step in resource development (Branets et al., 2009). These algorithms conventionally result in fine-scale descriptions of reservoir properties (porosity, permeability) on grids of tens of millions of cells (M. A. Christie, 1996).

A considerable important issue is the risk and uncertainty assessment of reservoir performance. The uncertainty can be gauged by simulating an ensemble of different geological realizations. This may require to run thousands of simulations to cover a wide range of parameter variation. It is however not computationally feasible or desirable to perform these simulations on the high-fidelity (fine-grid) model. Significantly upscaled models (i.e., the mapping of rock and fluid properties to a coarser resolution) are therefore required, where these models should ideally be even coarser than typical reservoir simulators, which can handle on the order of $10^5 - 10^6$ simulation cells (Durlofsky, 2005). In the presence of more complex physics, excessive upscaling may, however, result in non-satisfactory results, which necessitate the use of advanced algorithms and solvers to allow for higher resolution grids to be employed (Cusini et al., 2016).

Traditional Algebraic Multiscale techniques (Wang et al., 2014; Jenny et al., 2003), developed to solve the elliptic (or parabolic) pressure equation in sequentially coupled simulations, mainly focus on the pressure solution and often ignore the transport. However, in complex applications related to chemical and compositional EOR (Enhanced Oil Recovery), reservoir storage and geothermal industry, the number of conserved chemical species can be large which makes any improvement in transport solution quite significant for the performance and robustness of the simulation. A technique called *Adaptive Mesh Refinement* (AMR) provides an effective means for adapting the resolution of a model to solution requirements. This method is well developed in many areas of computational physics (e.g. fluid dynamics and solid mechanics) but is however not widely used for practical reservoir simulation (Karimi-Fard & Durlofsky, 2014).

In today's literature, several researchers have developed and proposed AMR procedures to capture the local nature of transport processes. Bahrainian and Dezfuli (2014) have developed a novel unstructured grid generation algorithm which considers the effect of geological features and well locations in the grid resolution. This strategy involves the definition and construction of the initial grid based on the geological model, geometry adaptation of geological features and grid resolution control. Trangenstein (2002) used the combination of high-resolution discretization methods with dynamically adaptive mesh refinement for a two-component single-phase model for miscible flooding. Pau et al. (2012) proposed an AMR algorithm for compressible two-phase flow in porous media. The method is implemented within a block structured adaptive mesh refinement framework which allows the grids to dynamically adapt to flow features and enables efficient parallelization of the algorithm. The coarse-scale permeability was obtained by averaging the fine-scale permeability. Similar techniques have been developed for compositional simulation (Sammon et al., 2003), thermal problems (Christensen et al., 2004), improved/enhanced oil recovery processes (Van Batenburg et al., 2011) and many more applications.

In this project, the aim was to develop a dynamic AMR scheme using an unstructured multi-level gridding framework, for general-purpose reservoir simulation. The focus lied particularly on thermal-reactive flow and transport formulation which are required for a wide range of subsurface applications. First, a fine-scale geological model was constructed accurately, approximating all reservoir heterogeneity. This model is represented by a list of volumes, depths and a connectivity list (Lim et al., 1995) describing each control volume. Next, a global flow-based upscaling was applied and an unstructured partitioning of the original grid was constructed as suggested in (Karimi-Fard & Durlofsky, 2014). This partitioning provides coarser levels of the original model each of which is also described by a list of volumes, depths and a connectivity list. A coarser connectivity list includes connections between control volumes at the given level as well as interconnections between the levels. Once the full hierarchy of levels is constructed, the simulation is started at the coarsest grid. Grid space refinement criterion is developed for particular applications. The multi-level connection list is reconstructed at each time step and used for the simulation. The proposed approach was implemented in *Delft Advanced Research Terra Simulator* (DARTS) which is based on Operator-Based Linearization techniques (Voskov, 2017).

The report outline is as follows; first, the methodology describing the relevant governing equations, the construction of the multi-level framework and the dynamic adaptivity scheme is presented. The approach was then tested on several applications of practical interests, including geothermal applications and reactive flow. The results of these test examples are presented in chapter 3, along with an error analysis. This is followed by concluding remarks on the methodology and the results of the tested applications. And finally, some recommendations for further research and development of the framework are discussed.

2

Methodology

2.1. Governing equations

General-purpose reservoir simulation is based on the solution of governing equations which describe mass and energy transfer of various species in the subsurface. The flow of mass and energy in a system with n_p phases and n_c components are described in this section. For this general-purpose thermal-compositional model, n_c component mass conservation equations and a single energy conservation equation need to be solved (Khait & Voskov, 2017). When chemical reactions occur in the system, an additional term describing n_k kinetic reactions is added on the right-hand side of the mass conservation equation. These governing relations are described as:

$$\frac{\partial}{\partial t}\left(\phi\sum_{p=1}^{n_p} x_{cp}\rho_p s_p\right) + \text{div}\sum_{p=1}^{n_p} x_{cp}\rho_p u_p + \sum_{p=1}^{n_p} x_{cp}\rho_p\tilde{q}_p = \sum_{k}^{n_k} v_{ck} r_k, \quad c = 1,\dots,n_c, \quad k = 1,\dots,n_k, \tag{2.1}$$

$$\frac{\partial}{\partial t}\left(\phi\sum_{p=1}^{n_p} \rho_p s_p U_p + (1-\phi)U_r\right) + \text{div}\sum_{p=1}^{n_p} h_p\rho_p\overrightarrow{u_p} + \text{div}(\kappa\nabla T) + \sum_{p=1}^{n_p} h_p\rho_p\tilde{q}_p = 0, \tag{2.2}$$

where t is the time, v_{ck} is the stoichiometric coefficient associated with kinetic reaction k, r_k is the rate of kinetic reaction k. The right-hand side of the mass conservation equation 2.1 is the kinetic term which describes reactions. It is set to zero when no chemical processes are involved in the system. The rest of the terms in the system can be described as functions of spatial coordinate $\boldsymbol{\xi}$ and/or physical state $\boldsymbol{\omega}$:

$\phi(\boldsymbol{\xi},\boldsymbol{\omega})$: porosity,
$x_{cp}(\boldsymbol{\omega})$: the mole fraction of component c in phase p,
$s_p(\boldsymbol{\omega})$: phase saturation,
$\rho_p(\boldsymbol{\omega})$: phase molar density,
$\overrightarrow{u_p}(\boldsymbol{\xi},\boldsymbol{\omega})$: phase velocity,
$\tilde{q}_p(\boldsymbol{\xi},\boldsymbol{\omega},\mathbf{u})$: source of phase p,
$\mathrm{U}_p(\boldsymbol{\xi})$: phase internal energy,
$\mathrm{U}_r(\boldsymbol{\xi})$: rock internal energy,
$h_p(\boldsymbol{\xi})$: phase enthalpy,
$\kappa(\boldsymbol{\xi},\boldsymbol{\omega})$: thermal conduction.

An exception is the phase source term, which is also dependent on $\mathbf{u}$ - well control variables.

The rock internal energy and thermal conduction are assumed to be spatially homogeneous for simplification of the problem, meaning that they are characterized as functions of the spatial coordinate $\boldsymbol{\omega}$ only. Phase flow velocity $\overrightarrow{u_p}$ is assumed to follow Darcy's law, expressed as:

$$\overrightarrow{u_p} = -\left(\mathbf{K}\frac{k_{rp}}{\mu_p}\left(\nabla P_p - \overrightarrow{\gamma}_p\nabla D\right)\right), \quad p = 1,\dots,n_p, \tag{2.3}$$

where

$\mathbf{K}(\boldsymbol{\xi})$: permeability tensor,
$k_{rp}(\boldsymbol{\omega})$: relative permeability of phase p,
$\mu_p(\boldsymbol{\omega})$: phase viscosity,
$P_p(\boldsymbol{\omega})$: pressure in phase p,
$\overrightarrow{\gamma}_p(\boldsymbol{\omega})$: gravity vector,
$D(\boldsymbol{\xi})$: depth (backward oriented).

The nonlinear unknowns in this system of equations are the pressure P, the overall compositions z_c of each component and the enthalpy h.

2.2. Modeling approach

In order to solve the governing equations 2.1 and 2.2, we apply a finite-volume discretization on a general unstructured mesh and perform a backward Euler approximation in time to both equations, where the phase velocities $\overrightarrow{u_p}$ are substituted by the Darcy relation (2.3):

$$\begin{aligned} V\left[\left(\phi\sum_{p=1}^{n_p} x_{cp}\rho_p s_p\right)^{n+1} - \left(\phi\sum_{p=1}^{n_p} x_{cp}\rho_p s_p\right)^{n}\right] - \Delta t\sum_l\left(\sum_{p=1}^{n_p} x_{cp}^l\rho_p^l\Gamma_p^l\Delta\psi^l\right) \\ + \Delta t\sum_{p=1}^{n_p}\rho_p x_{cp} q_p = V\Delta t\sum_k^{n_r} v_{ck} r_k, \quad c = 1,\dots,n_c, \quad k = 1,\dots,n_k \end{aligned}, \tag{2.4}$$

$$\begin{aligned} V\left[\left(\phi\sum_{p=1}^{n_p}\rho_p s_p U_p + (1-\phi)U_r\right)^{n+1} - \left(\phi\sum_{p=1}^{n_p}\rho_p s_p U_p + (1-\phi)U_r\right)^{n}\right] \\ - \Delta t\sum_l\left(\sum_{p=1}^{n_p} h_p^l\rho_p^l\Gamma_p^l\Delta\psi^l + \Gamma_c^l\Delta T^l\right) + \Delta t\sum_{p=1}^{n_p} h_p\rho_p q_p = 0. \end{aligned} \tag{2.5}$$

Here V is the control volume for which the system is being solved, $q_p = \tilde{q}_p V$ is a source of phase p, n is the previous time step whereas $n+1$ is the time step we want to solve for. Capillarity and gravity are neglected in these equations, and a Two-Point Flux Approximation (TPFA) with an upstream weighting is applied. $\Delta\psi^l$, the phase potential, therefore simply becomes the difference in pressure between blocks connected via interface l, while ΔT^l is the temperature difference between these blocks; $\Gamma_p^l = \Gamma^l k_{rp}^l/\mu_p^l$ is a phase transmissibility, where Γ^l is the constant geometrical part of the transmissibility (involving permeability and geometry of the control volume). Finally $\Gamma_c^l = \Gamma^l\kappa$ is the thermal (conductive) transmissibility (Khait & Voskov, 2017). This system of equations is solved for each mesh element in time, where the unknowns are the composition of the n_c components and the pressure for the mass conservation equation (2.4), and the pressure and enthalpy for the energy equation (2.5). To solve the system and these unknowns, it is therefore necessary to assign the relevant known properties to each grid cell, i.e. the volume, porosity and depth, and a connectivity list with corresponding transmissibility for each pair of connections.

In general-purpose reservoir simulation, the solving process requires the linearization of strongly nonlinear governing equations. In conventional reservoir simulators, a Newton-Raphson based method is typically used for the linearization, which solves on each nonlinear iteration a linear system of equations in the following form:

$$J(\boldsymbol{\omega}^n)(\boldsymbol{\omega}^{n+1} - \boldsymbol{\omega}^n) + r(\boldsymbol{\omega}^n) = 0, \tag{2.6}$$

where r is the residual and J is the Jacobian, which is the derivative of the residual with respect to primary nonlinear unknowns, defined at a nonlinear iteration n. In this work, we use a recently-developed approach called *Operator Based Linearization* (OBL). The main idea of OBL is to transform the discretized mass and energy conservation equations (2.4 and 2.5) to an operator form, where space-dependent ξ and state-dependent ω properties of governing equations are separated. This provides the opportunity to approximate the representation of the exact physics of a problem through the discretization of the state-dependent properties. The underlying methodology of OBL is explained in more detail in (Voskov, 2017) and (Khait & Voskov, 2017).

2.3. Multi-level grid generation

The adaptivity of the grid in the developed AMR scheme is based on a hierarchical representation. The simulation grid is composed of several predefined *levels* representing the same geological properties at different resolutions. The construction of such a multi-level grid is explained in this section.

We start with a fine-scale model (*static geological model*) which accurately represents all reservoir heterogeneity. This grid is defined as *level 0* and represents our finest level. The modeling grid is defined by a list of control volumes, depths, reservoir properties (all spatially distributed properties required to solve the discretized relations 2.4 and 2.5) for each mesh element, and a list of connectivity with corresponding transmissibility between neighbouring cells. I.e., each cell has a defined volume, depth and porosity, which are stored in an ordered list. Each cell has an assigned index number which associates it with corresponding properties. The connectivity list consists of two indexes, associated with cells adjacent to this connection, and the corresponding interface transmissibility. These concepts will be explained further in this section.

Next, *level 1* is defined, where control volumes are constructed by aggregating fine grid cells. Upscaling is applied to redefine volume, depth and reservoir properties at a coarser level. A connectivity list, with corresponding transmissibility, is constructed for this level and inter-level connections are defined in addition. Similarly, more levels of coarsening can be constructed. A control volume in grid-level n always consists of cells from grid-level $(n-1)$, resulting in a hierarchical relationship (Karimi-Fard & Durlofsky, 2014). The simulation grid is then obtained by combining control volumes from grids of different levels. A schematic representation of this procedure is illustrated in figure 2.1 below, where the different levels are shown, along with a potential configuration for the simulation grid.

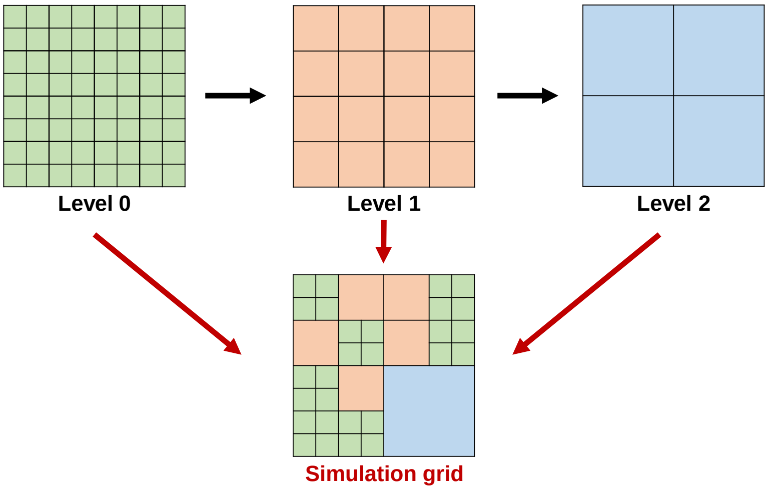


Figure 2.1: 2D Multi-level grid with three pre-constructed grids (levels) with an example simulation grid which is constructed by aggregating control volumes from different levels.

To solve the governing equations, we hence need a connectivity list (for connections within each level and also describing inter-level connections) with corresponding transmissibility, a list of volumes, a list of depths and other properties defined at each control volume within the hierarchical grid. The next sections will present the methodology behind the construction of the hierarchical grid and the determination of the required properties.

2.3.1. Cell aggregation

A mesh consists of a set of finite elements, each having vertices with allocated coordinates. To conduct cell aggregation, the centroid, defined as the arithmetic mean position of all the vertices in a given geometry, is first computed for each mesh element in the grid in question. Figure 2.2 below shows the scheme of an example 2D unstructured grid with Delauney triangulation, used to illustrate how cell aggregation is conducted. As can be seen, in this particular example, each cell has 3 vertices, and a centroid (represented in red) with coordinates x_c and y_c defined as $\left(\frac{x_1+x_2+x_3}{3}, \frac{y_1+y_2+y_3}{3}\right)$, where x_i and y_i are the coordinates of the vertices. Each mesh element has an assigned index number.

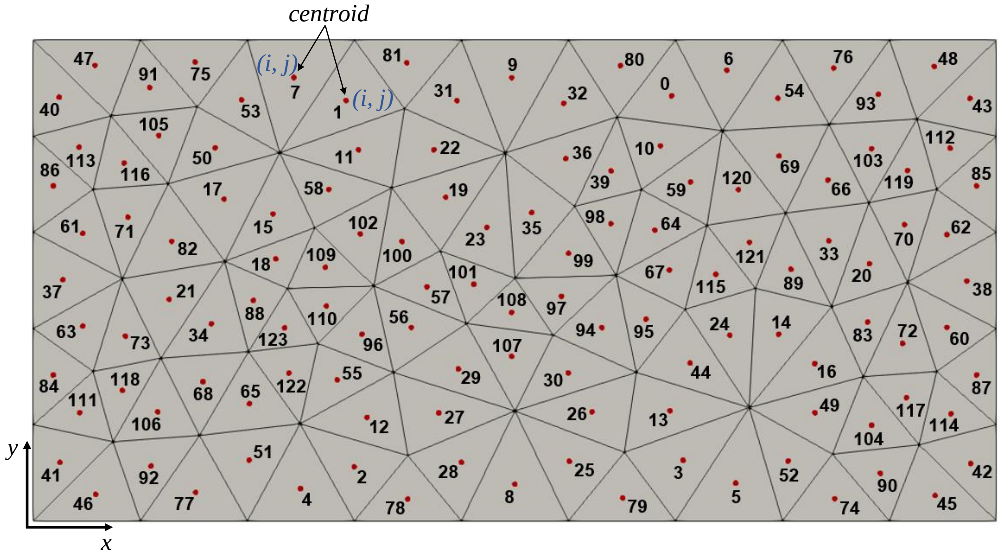


Figure 2.2: 2D unstructured grid with Delauney triangulation. The centroid of each mesh element is represented by a red point, with coordinates $(i,\ j)$. Each cell has an assigned index number.

Cell aggregation is then carried out by dividing the grid in the x- and y-direction (and in the z-direction for 3D models) into equidistant intervals. A predefined scaling factor is used in these directions, which results in intervals with Δx and Δy spacing in the x- and y-direction respectively. Each interval has coordinates $[i,\ i+\Delta x]$ in the x-direction and $[j,\ j+\Delta y]$ in the y-direction. Centroids of cells whose coordinates are within a given xy-area are aggregated to form one coarse cell. To check whether a fine cell f is within a given plane which will form coarse cell F, the following algorithm is implemented for the coordinates x_{cf} and y_{cf} of the centroid of fine cell f:

$$\text{if} \quad i \leqslant x_{cf} < i+\Delta x \quad \text{and} \quad j \leqslant y_{cf} < j+\Delta y, \quad \text{cell } f \in \text{cell } F. \tag{2.7}$$

Figure 2.3 below shows the range partitioning (illustrated by the white lines) for the above example (figure 2.2). The x- and y-range were divided in 5 and 3 equidistant intervals respectively. The yellow-highlighted 2D plane has range $[i,\ i+\Delta x]$ in the x-direction and $[j,\ j+\Delta y]$ in the y- direction. For this given example, all cell centroids whose x- and y- coordinates fall within this x- and y-range, are aggregated to form one coarse cell. I.e., cells 41, 46, 68, 77, 84, 92, 106, 111 and 118 form coarse cell 0.

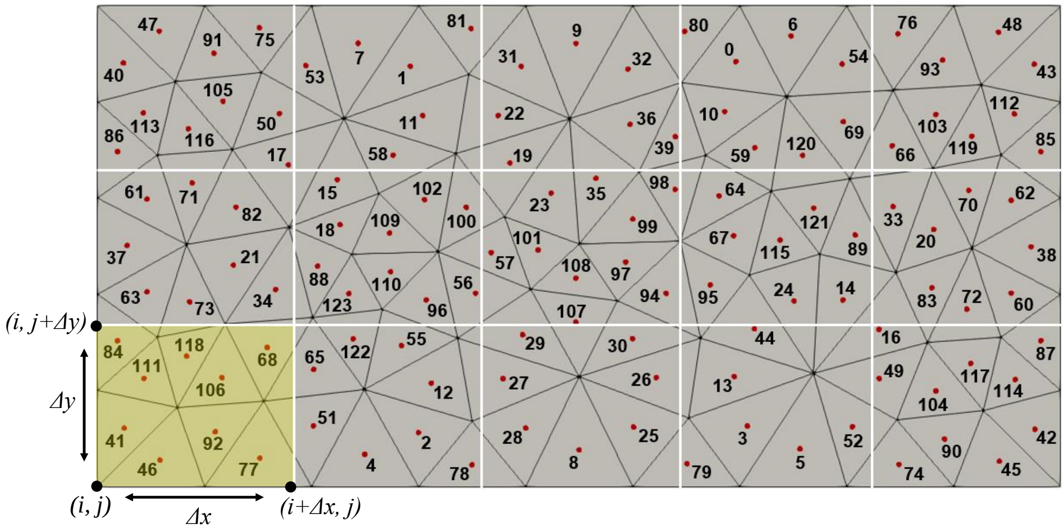


Figure 2.3: 2D unstructured grid from figure 2.2 with the range partitioning (represented by the white lines) in the x- and y-direction with Δx and Δy spacing respectively. Aggregation is carried out for cells whose centroid fall within a given x- ($[i : i+\Delta x]$) and y-range ($[j : j+\Delta y]$). E.g., all cells whose centroids are found within the yellow-highlighted 2D range are aggregated to form one coarse cell.

For the given 2D unstructured grid example above, the so-called *level 1* - i.e. the next level of coarsening - is shown in figure 2.3 below. The numbers represent the assigned indices of the newly constructed coarse cells. If one wants to construct an additional level, the same procedure can be followed with a larger x- and y-range partitioning, where grid cells of level 1 are aggregated to form level 2.

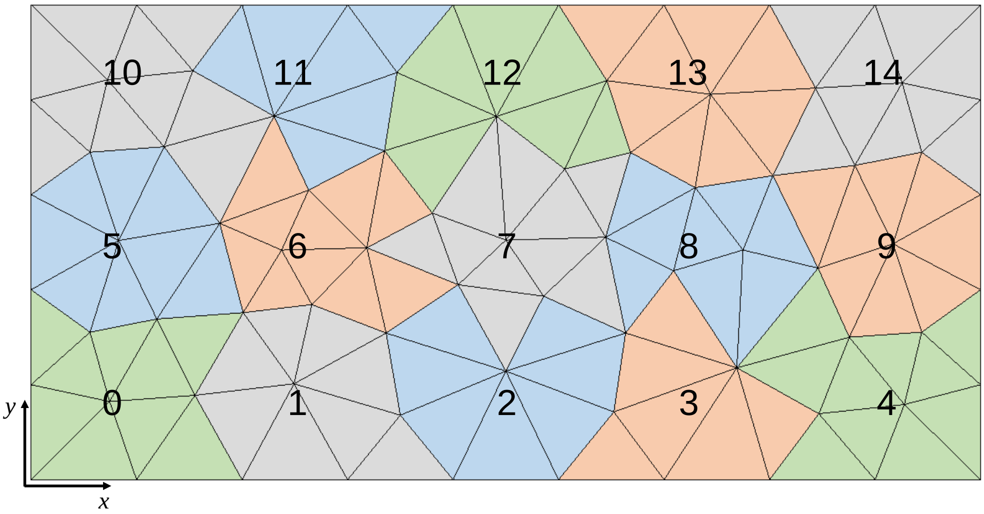


Figure 2.4: 2D unstructured grid of figure 2.2 after cell aggregation. This grid represents the next level of coarsening of the finest grid: level 1.

For further steps into the construction of the levels, a list - *"fines in coarse"* - is constructed where the corresponding indices of the aggregated fine cells are listed for each coarse cell. Table 2.1 below tabulates this list for the example above (figures 2.2 to 2.4). This type of list is generated for each coarse level (level > 0) in the hierarchical grid. These lists are useful for the construction of the cell properties (e.g. volume, porosity) of the coarse levels, where the cell data from the fine level is needed during upscaling.

Table 2.1: "fines_in_coarse" list of the example 2D unstructured grid of figures 2.2 - 2.4. This list describes for each coarse cell, the aggregated fine cells of level $n-1$ to form the coarse cell of level n.

Coarse cell	**Fine cells**
0	41, 46, 68, 77, 84, 92, 106, 111, 118
1	2, 4, 12, 51, 55, 65, 78, 122
2	8, 25, 26, 27, 28, 29, 30
3	3, 5, 13, 44, 52, 79
4	16, 42, 45, 49, 74, 87, 90, 104, 114, 117
5	21, 34, 37, 61, 63, 71, 73, 82
6	15, 18, 56, 88, 96, 100, 102, 109, 110, 123
7	23, 35, 57, 94, 97, 98, 99, 101, 107, 108
8	14, 24, 64, 67, 89, 95, 115, 121
9	20, 33, 38, 60, 62, 70, 72, 83
10	17, 40, 47, 50, 75, 86, 91, 105, 113, 116
11	1, 7, 11, 53, 58, 81
12	9, 19, 22, 31, 32, 36, 39
13	0, 6, 10, 54, 59, 69, 80, 120
14	43, 48, 66, 76, 85, 93, 103, 112, 119

Note that cell aggregation can also be conducted while taking care of highlighting geological features (e.g. fractures) and different facies in the model. For example, cell aggregation can be conducted by grouping domains with the same facies together into one coarse cell, or, in fractured reservoirs, by aggregating cells by isobar contours similar to (Karimi-Fard & Durlofsky, 2014).

2.3.2. Connectivity list

The proposed AMR technique uses the Finite Volume Method (FVM) for discretization. The implementation of the finite volume discretization method to the mass conservation equation 2.1 requires the evaluation of the flow between two adjacent control volumes in terms of the cell pressures. Using a Two-Point Flux Approximation (TPFA), the flow rate is defined as:

$$Q_{ij} = \Gamma_p^{ij}(P_i - P_j), \tag{2.8}$$

where:

Q_{ij} : flow rate between cells i and j,
Γ_p^{ij} : phase transmissibility between cells i and j,
P_i : pressure of cell i,
P_j : pressure of cell j.

A schematic representation of the TPFA is shown in the figure below, where pressure is defined at the cell center, and flow rate Q_{ij} and phase transmissibility Γ_p^{ij} are defined at the interface between the two cells.

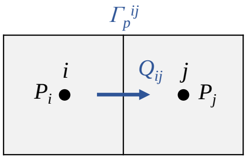


Figure 2.5: Flux representation based on the two-point flux approximation for two neighbouring grid cells i and j with pressures P_i and P_j defined at the cell centers and transmissibility Γ_{ij} defined at the interface.

Similarly, the heat flux between two adjacent control volumes is expressed in terms of thermal transmissibility Γ_c and is, also using a TPFA, defined as:

$$Q^h_{ij} = \Gamma_c^{ij}(T_i - T_j), \tag{2.9}$$

where Γ_c^{ij} is the thermal transmissibility at interface ij, T_i and T_j are the temperatures of cell i and j respectively, and Q^h_{ij} is the heat flux at interface ij.

To evaluate the flux between two adjacent control volumes, it is necessary to know and store existing neighbouring blocks within the mesh. A so-called *connectivity list* is therefore constructed where for each interface between two control volumes, the indices of these cells are listed together with the transmissibility value (Lim et al., 1995). The result is a list with all connection pairs present in the grid. A few important points to be noted are:

- Each connection consists of only two elements,
- The connection pairs are not repetitive,
- No-flow boundaries imply the absence of connections and are hence not listed in the connectivity list.

The figure below shows a simple example of a 2D Cartesian structured grid, with corresponding cell indexing.

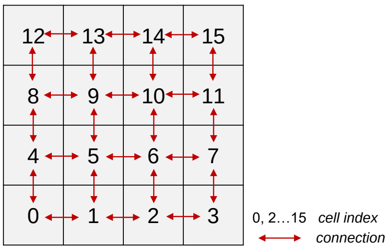


Figure 2.6: 4x4 Cartesian grid denoting cell indexing and showing neighbouring connections. Indexing is based on a Cartesian structured mesh for simplicity.

As can be seen in figure 2.6 above, cell 0 is connected to cell 1 and 4. Similarly, cell 1 is connected to cells 0, 2 and 5. Evaluating each control volume results in a list representing all dual connections within the grid. Table 2.2 below shows the connectivity list for the above example (figure 2.6). The list is expressed as two columns, cell i and cell j, where each row represents a connection pair.

Table 2.2: Connectivity list of the example grid from figure 2.6.

	Cell *i*	**Cell *j***
Dual connections	0	1
	0	4
	1	2
	1	5
	2	3
	2	6
	3	7
	4	5
	4	8
	5	6
	5	9
	6	7
	6	10
	7	11
	8	9
	8	12
	9	10
	9	13
	10	11
	10	14
	11	15
	12	13
	13	14
	14	15

In the proposed AMR scheme, the connectivity list of each level is determined systematically. Each mesh element consists of a set of vertices $\mathbf{x}$. E.g. a triangular mesh element comprises 3 vertices, and a Cartesian grid comprises 4 vertices. These vertices are numbered uniquely. The vertices $\mathbf{x}$ comprised in a cell i are stored in a list; this is done for each mesh element in level 0. To determine whether two control volumes i and j are adjacent, we take the intersection of both sets of vertices. That is:

$$\{\mathbf{x}|\mathbf{x} \in i\} \cap \{\mathbf{x}|\mathbf{x} \in j\}. \tag{2.10}$$

If the intersection size, in the case of a 2D grid, is equal to 2, both cells have a line in common, indicating that they have a common interface and are hence connected.

Each geometry has a different criterion. For 2D shaped mesh elements, the interface is a line; for 3D shaped cells, the interface is a plane. Hence the criterion is that the intersection length should equal 2 for 2D shapes and 3 or more for 3D shapes. This methodology is applied to the finest level of refinement - level 0. The result is a connectivity list representing all the unique connection pairs within level 0. The interface area is subsequently computed (and stored for transmissibility computation in further steps) for each connection, using the coordinates of the intersecting vertices.

The next step is to store for each cell i, connection pairs (interfaces) which consist of cell i, i.e. the interface which consists of a face of cell i. This is easily done by simply storing all the connection pairs which have cell i listed. Table 2.3 below lists the methodology for the example above (figure 2.6).

Table 2.3: Interfaces l contained in each cell i for level 0. The interfaces are expressed as connection pairs, describing an interface.

Cell *i*	**Faces**
0	(0, 1), (0, 4)
1	(0, 1), (1, 2), (1, 5)
2	(1, 2), (2, 3), (2, 6)
3	(2, 3), (3, 7)
4	(0, 4), (4, 5), (4, 8)
5	(1, 5), (4, 5), (5, 6), (5, 9)
6	(2, 6), (5, 6), (6, 7), (6, 10)
7	(4, 7), (6, 7), (7, 8)
8	(4, 8), (8, 9), (8, 12)
9	(5, 9), (8, 9), (9, 10), (9, 13)
10	(6, 10), (9, 10), (10, 11), (10, 14)
11	(7, 11), (10, 11), (11, 15)
12	(8, 12), (12, 13)
13	(9, 13), (13, 14)
14	(10, 14), (13, 14), (14, 15)
15	(11, 15), (14, 15)

For coarser levels (level > 0), a list with corresponding fine connection pairs is also constructed, where all the connection pairs of the fine grid cells i contained in coarse cell I are stored. This is done for each coarse cell. This way, we can now easily determine the coarse level connectivity list. This is done similarly as for level 0, however, now the common fine grid faces are determined instead of the common vertices. To illustrate the methodology, the simple example grid from figure 2.6 is taken. Cell aggregation is conducted, which results in four coarse grids as depicted in the figure below, where the grey cell is coarse cell 0, the green cell is coarse cell 1, the blue cell coarse cell 2, and the pink cell coarse cell 3:

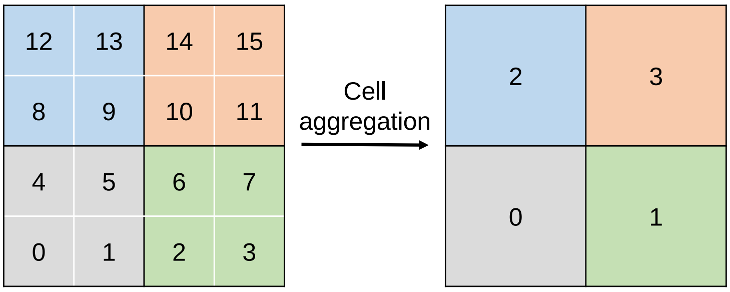


Figure 2.7: 4x4 Cartesian grid showing cell aggregation, resulting in four coarse grid cells.

Next, we construct hierarchical connection lists. First, the fine-scale interfaces (represented by connection pairs) are stored in a list for each coarse block. The connections which consists of two fine cells that are both found within a given coarse cell I are unaccounted for, as they form inner interfaces, whereas we are interested in the outer faces. For the example above, this results in the following list tabulated in table 2.4 below:

Table 2.4: List of faces (corresponding connection pairs) for each coarse cell in the coarse grid.

Coarse cell *I*	**Faces**
0	(1, 2), (4, 8), (5, 6), (5, 9)
1	(1, 2), (5, 6), (6, 10), (7, 11)
2	(4, 8), (5, 9), (9, 10), (13, 14)
3	(6, 10), (7, 11), (9, 10), (13, 14)

Next, the common faces between each coarse cell are determined. This is implemented by evaluating the intersection between the set of faces $\mathbf{f}$ belonging to coarse cell I and the set of faces $\mathbf{f}$ forming coarse cell J.

This is expressed as:

$$\{\mathbf{f}|\mathbf{f} \in I\} \cap \{\mathbf{f}|\mathbf{f} \in J\}. \tag{2.11}$$

If a given coarse cell I has a or multiple common faces f with another coarse cell J within the grid, these two cells form neighbouring blocks. For further steps such as the transmissibility computation, the area of the connecting interface is stored, which is here expressed at the sum of the intersecting faces.

For inter-level connections, a similar method is implemented. For each coarse cell I in level n, the intersection of its set of faces $\mathbf{f}$ with the set of faces $\mathbf{f}$ of a given fine cell i is determined. This operation is conducted for every fine cell i in level $(n-1)$, except for the fine cells comprised in the evaluated coarse cell I $(i \in I)$. This is expressed mathematically as follows:

$$\{\mathbf{f}|\mathbf{f} \in I\} \cap \{\mathbf{f}|\mathbf{f} \in i\}, \quad \text{where} \quad \{i|i \notin I\} \tag{2.12}$$

Similarly, if a given coarse cell I has a common face with a fine cell i, the two cells are connected. This procedure is applied between all levels n and $(n-1)$. The result is a list of connections within level 0, a list of connections for each level n, and an inter-level connectivity list, each having a corresponding interface area stored.

2.3.3. Transmissibility computation

The transmissibility is defined as the ratio between the rate and pressure gradient applied at a given interface. It is a key parameter to compute the flux between two neighbouring cells. In this project, the AMR method is implemented for unstructured grids of any geometry. The definition of the transmissibility for unstructured grids is expressed as:

$$\Gamma_p^{12} = \Gamma^{12}\lambda \quad \text{with} \quad \Gamma^{12} = \left(\frac{\alpha_1\alpha_2}{\alpha_1+\alpha_2}\right) \quad \text{and} \quad \alpha_i = A\frac{k_i}{D_i} n \cdot d_i, \tag{2.13}$$

where:

Γ_p^{12} : transmissibility between cells 1 and 2,
Γ^{12} : constant geometrical part of the transmissibility,
λ : mobility of a given phase p,
A : interface area,
k_i : permeability of cell i,
D_i : distance between centroid of cell i to interface area A,
n : unit vector normal to the interface,
d_i : unit vector along the line joining centroid of cell i to the center of interface A.

Here, the directional permeability of each cell is expressed as the magnitude of the cell's $[k_x, k_y, k_x]$ coordinates multiplied by the unit vector d_i.

Figure 2.8 below is a schematic representation of two unstructured grid cells with relevant parameters to compute the transmissibility across the interface.

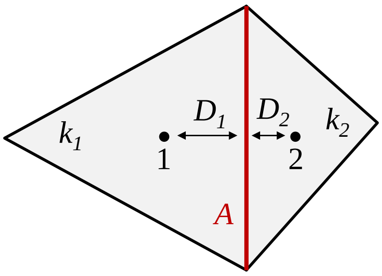


Figure 2.8: Transmissibility representation for two neighbouring unstructured cells 1 and 2 with permeability k_1 and k_2 and distances D_1 and D_2 to the interface separating the two control volumes with interface area A.

To solve the mass conservation equation (eq. 2.1), the flow rate must be computed for the interface of every neighbouring cells. It is therefore necessary to compute the transmissibility for each dual connection listed in the connectivity list. The result is a list consisting of all connections, with their corresponding transmissibility. This methodology is applied at the finest level of refinement, level 0.

For thermal problems, another type of transmissibility Γ_c^l must be computed to approximate thermal conductive flux in the energy equation 2.2. Since thermal rock conduction is not as heterogeneous as permeability, the thermal transmissibility is defined as the geometric coefficient, that is, the area of the interface l divided by the sum of the distances D_1 and D_2 from centroids to interface l, multiplied by the average conduction κ_{12}. For the example illustrated in figure 2.8, the thermal transmissibility is expressed as:

$$\Gamma_c^{12} = \kappa_{12} \frac{A}{D_1 + D_2}. \tag{2.14}$$

2.3.4. Upscaling

As mentioned earlier, level 0 is represented by a list of volumes, depths and reservoir properties. I.e., each mesh element has a defined volume, depth and reservoir properties, which are derived from the static geological model. Once the hierarchical grid is constructed, all cell properties must be redefined for the coarser levels (level > 0). This is done by upscaling the properties of the corresponding fine grid cells.

The volume is upscaled by simply summing the volumes of the aggregated fine grid cells v_i;

$$V_I = \sum_{i \in I} v_i. \tag{2.15}$$

If depth upscaling is needed, i.e. if cell aggregation is carried out in the z-direction, it is conducted by simply computing the z-coordinate value of the coarse cell centroid (average of the vertices of the fine-grid cells i comprised in a given coarse cell I) as an upscaled depth value.

The porosity, thermal conductivity, and rock heat capacity are upscaled using a volumetric averaging. For example, the sum of the porosity ϕ_i multiplied by the corresponding cell volume v_i of each fine cell i is taken over the total volume of the coarse cell V_I;

$$\bar{\phi}_I = \frac{1}{V_I} \sum_{i \in I} v_i \phi_i. \tag{2.16}$$

In this study, for the upscaling of permeability, we use the flow-based upscaling technique developed by (Karimi-Fard et al., 2006; Gong et al., 2008; Karimi-Fard & Durlofsky, 2012). This technique uses the pressure solution when the system has reached steady-state to compute the flow across each interface. The transmissibility can then be derived through rearranging the flow equation 2.8. These approaches can be applied to unstructured coarse grids with generally-shaped control volumes (Karimi-Fard & Durlofsky, 2014). The coarsening technique defines the coarse transmissibility Γ_p^{IJ} between two adjacent control volumes I and J. This is expressed as:

$$\Gamma_p^{IJ} = \left| \frac{Q_{IJ}}{P_I - P_J} \right|. \tag{2.17}$$

The coarse-grid average pressures P_I and P_J, and the coarse-grid flow rate Q_{IJ}, are computed using a fine-grid pressure solution. These quantities are given by:

$$P_I = \frac{1}{V_I} \sum_{i \in I} v_i p_i, \quad P_J = \frac{1}{V_J} \sum_{j \in J} v_j p_j, \quad Q_{IJ} = \sum_{(i \in I, j \in J)} Q_{ij} = \sum_{(i \in I, j \in J)} \Gamma_p^{ij} (p_i - p_j), \tag{2.18}$$

where p_i and p_j define the fine-scale pressures in the corresponding coarse blocks. In the flow rate expression Q_{ij}, ij indicates the interface between fine cells i and j and Γ_p^{ij} denotes the transmissibility for this interface. This ij interface comprises a portion of the interface between coarse blocks I and J. To find the fine connections consisting of the interface between coarse blocks I and J, i.e. the interface IJ, the fine connectivity list is looped where dual connections that consist of one fine cell in coarse block I and its neighbour comprised in coarse block J are stored. Then equation 2.18 is applied to these connections.

For inter-level connections, a similar approach is followed. Equation 2.18 is used with however P_I being the fine-scale pressure p_i of the fine grid in question, and P_J remaining the coarse-scale pressure, computed as outlined in equation 2.18. This procedure is conducted for each inter-level connection found within the hierarchical grid.

For thermal problems, a similar method can be implemented, but is however not computationally efficient as temperature takes significantly longer to reach a steady state. We therefore use equation 2.14 to compute the upscaled thermal transmissibility, where the area is expressed as the sum of the fine-scale faces which compose interface IJ, and the distances D_I and D_J represent the distances between the cell centroid and the centroid of the coarse interface.

2.3.5. Global indexing

Once all needed parameters at every hierarchical level are evaluated, which include cell properties, a connectivity list with associated transmissibility for each level and between levels, it is necessary to combine the levels in order to form a global hierarchical set of grids. To combine the levels, it is however necessary to assign a unique indexing to each and every mesh element contained in the multi-level grid. This is where global indexing plays a role. For convenience, indexing is ordered starting from the coarsest level. An example of global indexing is shown in the figure below.

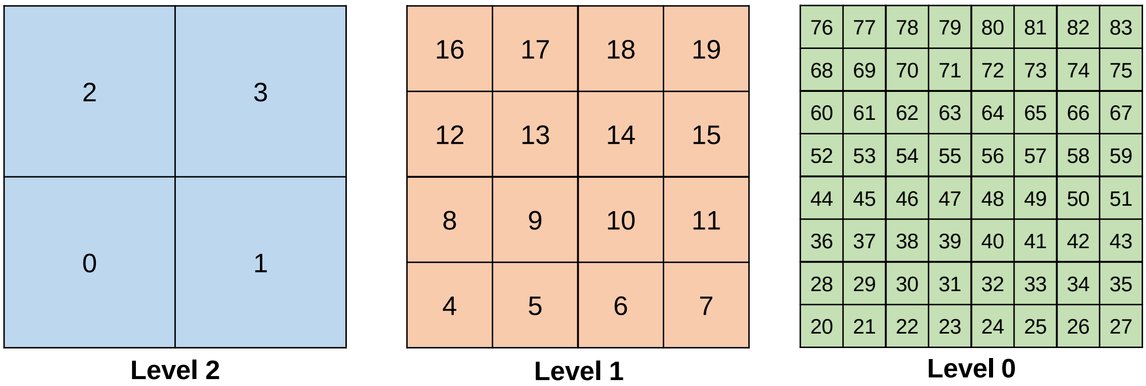


Figure 2.9: Example of a Cartesian structured 3-level hierarchical grid with global indexing. Numbering starts at the coarsest level and finishes at the finest level of refinement.

The procedure used to assign global indexing is to simply add the number of cells of the previous level(s) to the current level. E.g., for level 1 from the figure above, numbering starts at the total number of cells of level 2 n_2; for level 0, numbering starts at $n_2 + n_1$. This procedure is applied to the bookkeeping lists such as *"fines in coarse"* and to the connectivity list of the corresponding levels. After the global indexing is applied to the connectivity lists, the connectivity list at each level and the inter-level connection lists are combined into one list. This is conducted by concatenating these lists to form one list describing all existing connections within the hierarchical grid. Regarding the list of cell properties, global indexing is applied by simply concatenating the lists together in the right order, i.e. from the coarsest level to the finest level. This way, indexing is done in the same order as the global indexing. The result is a global list of volumes, depths and relevant reservoir properties describing each mesh element within the hierarchical grid.

Having constructed the hierarchical grid and assigned it global indexing, the pre-processing stage is complete and the simulation with dynamic adaptivity can be performed.

2.4. Dynamic adaptivity framework

Once the hierarchical grid of predefined levels is constructed, the pre-processing stage is ready and hence simulation process can start. This section describes the procedure undertaken for implementation of dynamic refinement and/or coarsening.

To find the solution to the governing equations 2.1 and 2.2, we need to solve for the solution variables (pressure and composition for the mass conservation equation and pressure and enthalpy for the energy equation). To do so, as mentioned earlier, we need the known variables for each and every mesh element within the grid; that is, the volume, depth and reservoir properties. Moreover, to solve for the flux term, we also need a connectivity list with corresponding transmissibility. The model implemented in DARTS therefore requires as input a list of volumes, depths, reservoir properties describing every grid cell within the simulation grid, and a list of connections and corresponding transmissiblity.

Furthermore, as not all cells present in the hierarchical grid are within the simulation grid, we hence require a list of active blocks, which is a list where the indices (global indices) of the active cells are stored.

To determine whether grid adaptivity is necessary, we define refinement and coarsening criteria, which are dependent on the application used. In this study, we adopted an approach where the difference in solution variable is analysed between neighbouring blocks. Therefore, the difference in the solution variable of interest X is computed between each pair of cells active in the simulation grid. If this difference is higher than a given threshold, both neighbouring blocks are refined. For the coarsening of a set of fine cells, belonging to a given coarse cell, the difference between all the corresponding fine cells and their neighbouring cells is computed; if each and every one of these connections have a difference in solution variable below a given threshold, the fine cells are coarsened to the next consecutive level.

For cells marked for refinement, the corresponding fine cells from the level below are added to the list of active blocks, which is used for implementation of the next time step, while the indices of the coarse cells in question are suppressed. Similarly, the cells marked for coarsening are suppressed from the active cells, and the corresponding coarse blocks are added. Figure 2.1 below shows an example of a two-level hierarchical grid. The current time step simulation grid is represented on the bottom left. After a check for adaptivity was conducted, cells 1 and 2 were marked for refinement. Hence as explained above, the cell indices 1 and 2 are suppressed from the list of active blocks, and their corresponding fine cell indices are added (6, 7, 10 and 11 for coarse cell 1, and 12, 13, 16 and 17 for coarse cell 2). The scheme at the bottom right of the figure shows the simulation grid which will be used for the next time step. Cell adaptivity results in an unstructured indexing as shown in the figure below.

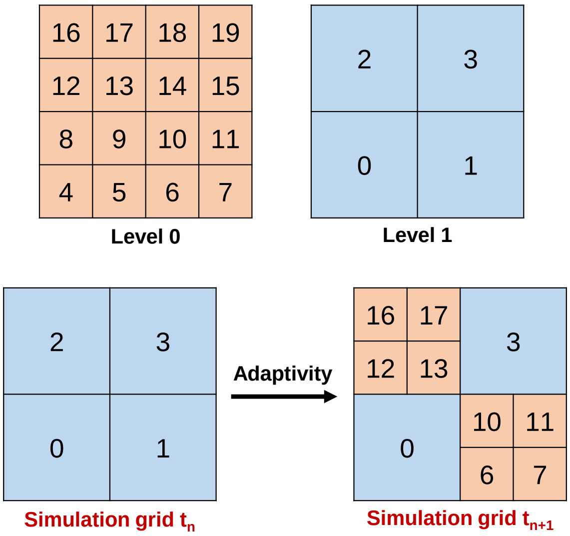


Figure 2.10: Schematic representation of a 2-level hierarchical grid, with illustrated the adaptivity procedure and the redefinition of the active blocks for the simulation grid of the next time step t_{n+1}.

Once the simulation grid is redefined and the list of active cells is updated, the connectivity list and corresponding transmissibility must be redefined. This is done by copying the list of connections for the whole hierarchical grid, where only the connections and corresponding transmissibility involving the active cells are kept, while connections involving non-active cells and their corresponding transmissibility are suppressed. Similarly, the same holds for the list of volume, depth and relevant properties. Only the cell properties of the active blocks are stored.

For computation of the next time step solution X_{n+1}, the solution of the previous time step X_n is required (see equations 2.4 and 2.5). However, X_n doesn't have the same grid configuration as the next time step t_{n+1}. It is therefore necessary to convert the grid of solution X_n to the same configuration as the simulation grid at t_{n+1}. To do so, we use simple mapping techniques. A prolongation operator is firstly used to redefine the solution variable X at each cell of the finest level of refinement (level 0). A so-called *constant prolongation* is implemented; i.e., all sub-domain values X_i are set to the coarse value solution variable X_I:

$$X_i = X_I, \quad \forall i \in I. \tag{2.19}$$

Subsequently, restriction to the new simulation grid is conducted on the prolongated solution; i.e., for cells already at the finest level, the solution stays the same; when several control volumes are grouped into a single coarser control volume, the coarse value X_I is set to the volume-weighted average of all sub-domain values X_i (Karimi-Fard & Durlofsky, 2014):

$$X_I = \frac{1}{V_I} \sum_{i \in I} v_i X_i. \tag{2.20}$$

A schematic representation of this procedure for the 2-level hierarchical grid and for the new simulation grid of figure 2.10 (t_{n+1}) is shown in figure 2.11 below.

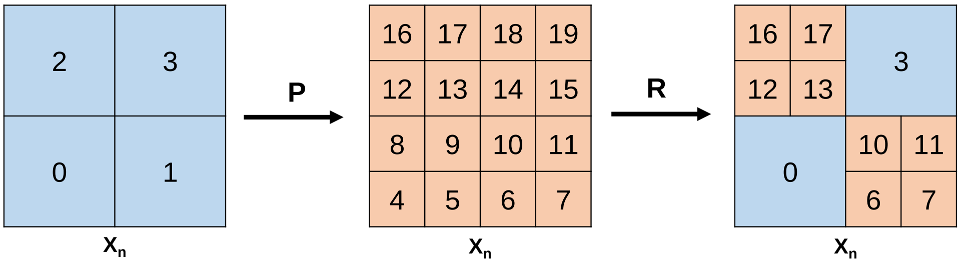


Figure 2.11: Schematic representation of the prolongation and restriction operation for the example in figure 2.10 above.

The model, however, necessitates sequential numbering for mesh generation. It can be seen in figure 2.10 that indexing is non-consecutive when grid adaptivity is applied. This is where local indexing comes in play. That is, the active blocks indices are re-numbered in a sequential order to prevent undefined indices in the mesh. The figure below shows this procedure for the problem described in figure 2.10.

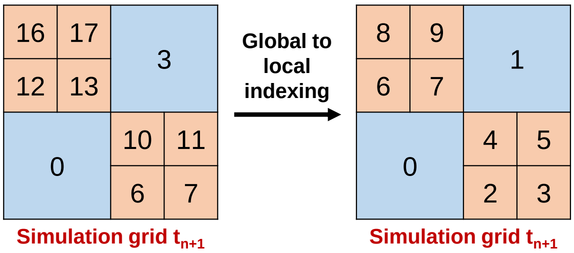


Figure 2.12: Example of the global to local index conversion for the simulation grid of time t_{n+1} for the problem illustrated in figure 2.10.

This new local indexing is assigned to the list of cell properties (volume, depth, porosity...etc).

To keep track of the global indexing of each active cell for future steps in the procedure, a so-called *Global to Local* array is constructed. This consists of a list, with size corresponding to the total number of blocks in the hierarchical grid, where the index represents the global indexing and the value represents the local indexing for the current time step. Blocks that are not active are given a value of -1. This array is used to convert the global connectivity list to local numbering. This array is redefined at each time step. Table 2.5 below lists the *Global to Local* array for the simulation grid shown in figure 2.12.

Table 2.5: Global to local array with listed the local indexing for a given grid cell in the new simulation grid (having a global index assigned).

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

In the synthetic examples used to illustrate the performance of the AMR framework, the first time step simulation is started at the coarsest level. For improved accuracy, the cells containing the wells are kept at the finest level of refinement, level 0. We therefore construct an active list containing the indices of the well blocks, and the remaining indices of the coarsest level. Global to local indexing is then applied in order to avoid non-assigned indices. When primary variables are solved, it is important to convert the local indexing back to the global indexing (using *Global to Local* array) in order to perform the adaptivity check and to redefine the active list and new simulation grid.

The procedure described above, which redefines the grid configuration for the next simulation, is repeated at each time t. Both the pre-processing stage, which consists of the construction of the hierarchical grid, along with the dynamic refinement framework are summarized in the flow chart shown in figure 2.13 below.

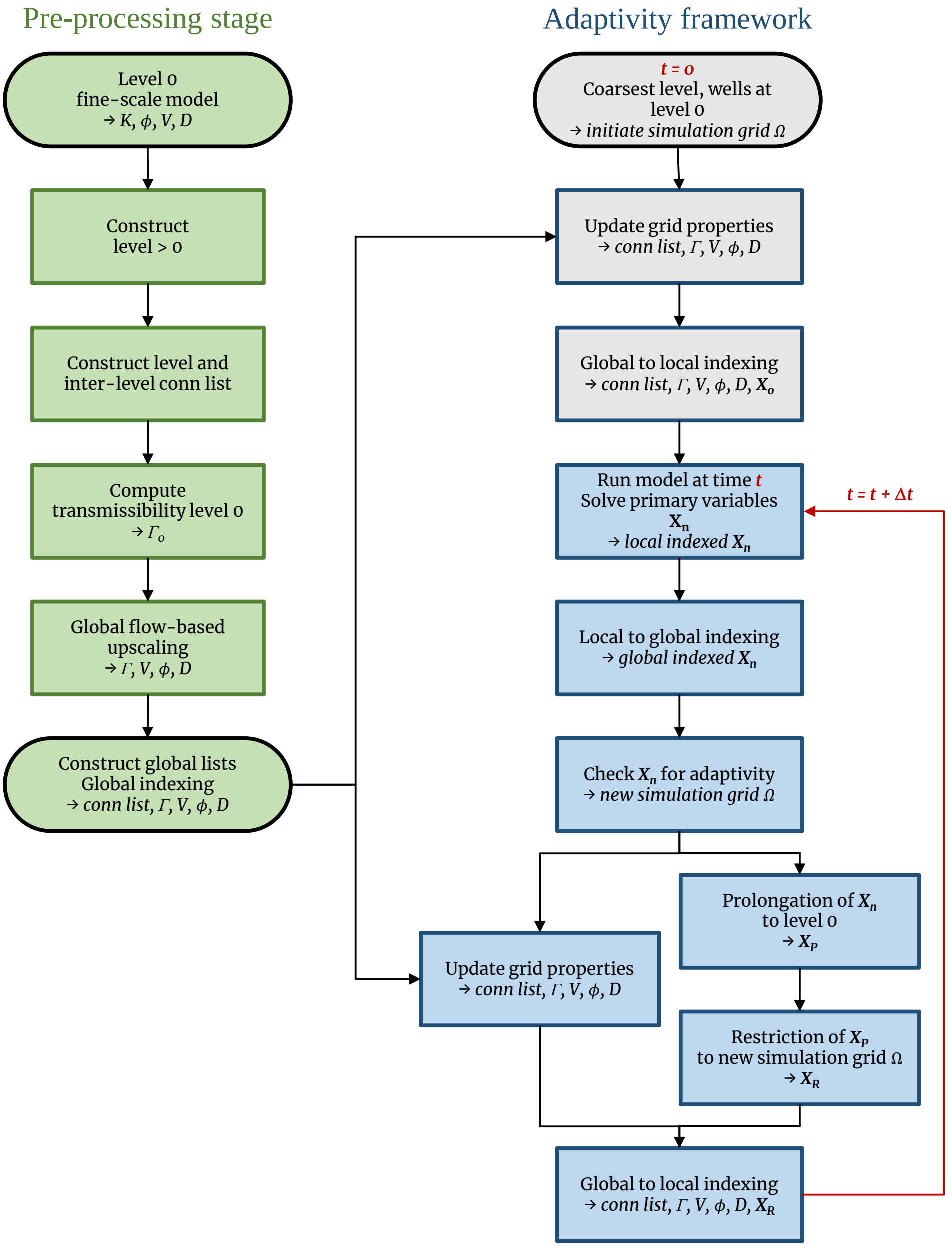


Figure 2.13: Flow chart describing the steps in the pre-processing stage, i.e. the construction of the hierarchical grid; and describing the adaptivity framework.

3

Applications

The performance of the developed AMR method is illustrated for several applications in this chapter. The applications, tested in this study, are a production of geothermal energy and a reactive flow of acid in carbonate reservoirs. For all cases, we consider a single injector single producer configuration. This section presents an analysis based on comparison between the solution accuracy among the fine, coarse and AMR grid.

3.1. Geothermal reservoirs

Geothermal technology has recently received substantial attention as an alternative source of energy. However, geothermal production systems have a relatively low return on investment, where uncertainties related to lack of detailed information about subsurface formations can significantly affect the quantification of the economic planning and feasibility of geothermal projects (Willems, 2017). It is therefore important to reduce the uncertainty and produce a high accuracy solution while keeping the computational costs low. Geothermal systems therefore represent a good candidate for implementation of our AMR framework since it keeps the accuracy of simulation process close to the fine-scale while the performance is close to coarse-scale models.

Simulation of geothermal reservoirs implicates the solution of both mass 2.1 and energy 2.2 conservation equations where pressure and enthalpy are the solution variables. We are mostly interested in the accurate prediction of the temperature displacement front and resulting thermal breakthrough time. Dynamic adaptivity will be illustrated for four synthetic geothermal examples:

- A homogeneous reservoir with unstructured meshing,
- A simple sugar-cube-like fluvial model with shale blocks,
- A heterogeneous fluvial system from (Shetty et al., 2017) with low net-to-gross ratio,
- And finally, several layers of a highly heterogeneous SPE10 reservoir.

In *DARTS*, the enthalpy is used as nonlinear unknown instead of the temperature. The adaptivity criteria are therefore applied to the enthalpy solution where the difference in enthalpy between two adjacent control volumes is analysed. This is done for each pair of connection within the simulation grid. Here, we applied the following adaptivity criteria:

$$\begin{cases} \text{if } \Delta h_{ij} > 70\ kJ, & \text{mark cells i and j for refinement,} \\ \text{if } \Delta h_l < 40\ kJ, \quad \forall l \in I, & \text{mark cells } \{\forall i \in I\} \text{ for coarsening.} \end{cases} \tag{3.1}$$

3.1.1. Homogeneous model

The first model is a simple 2D homogeneous reservoir (constant permeability) with unstructured triangular mesh. As mentioned before, we consider a single injector (I) and a single producer (P) configuration. A two-level hierarchical grid is used, with 1420 cells in level 0 and 75 cells in level 1. Figure 3.1 below illustrates both levels, along with the permeability field (constant permeability of 2000 mD), and the well locations.

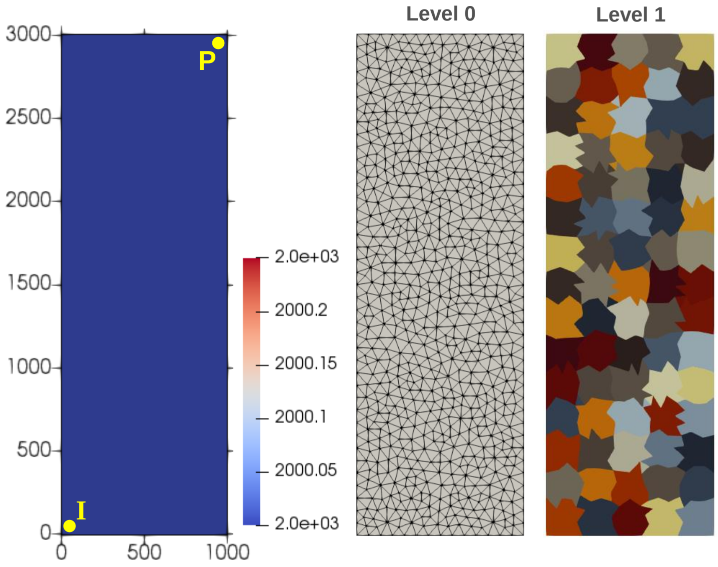


Figure 3.1: Hierarchical grid of the unstructured homogeneous model. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement, with unstructured gridding; right figure: coarser level, level 1 where each color represents a coarse cell.

The level 1 is illustrated above where each color represents a coarse cell. As can be seen, cell aggregation has been conducted by dividing the x- and y- axes into 5 and 15 equidistant intervals. The cells at the well locations are kept fine at all times. The simulation was conducted for a period of 5500 days. The temperature solution at three different times is shown in figure 3.2 below. Figure (a) represents the temperature solution at fine scale, figure (b) the solution on the AMR grid, figure (c) shows the coarse-scale solution, and figure (d) shows the node distribution for the AMR simulation run.

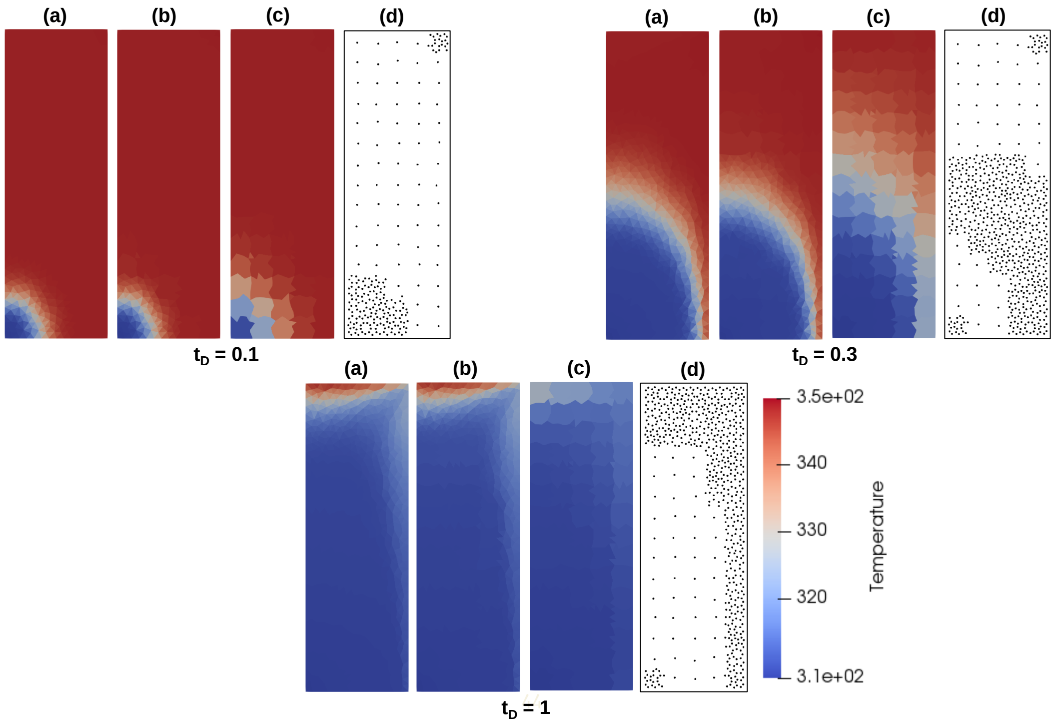


Figure 3.2: Temperature solution of the homogeneous reservoir with unstructured gridding at $t_D = 0.01$, 0.3 and 1. (a) represents the fine-scale solution; (b) represents the AMR solution; (c) is the coarse-scale solution; (d) is the node distribution of the AMR grid.

As can be seen, the solution on the AMR grid demonstrates a particularly good match with the fine-scale solution. Moreover, the node distribution shows high concentration along the front and at the well locations, and low concentration behind and ahead of the front, where no significant changes are observed. This considerably lowers the computational time as compared to running the fine-scale model. The coarse-scale solution differs notably from the AMR and fine-scale solution, with a faster cold front propagation at the coarse grid which is more pronounced in comparison at late times $t_D = 0.3$ and 1.

To describe the differences quantitatively, an error analysis was conducted, where the error of both AMR and coarse solution are computed relative to the fine-scale solution. Both the L2 norm and L-infinity norm were calculated for each time step throughout the simulation. Figure 3.3 below shows both norms for the error between the AMR result and the fine-scale solution, and the error between the coarse-scale and fine-scale solution. As can be seen, the relative error of the AMR solution is significantly lower than the coarse solution in both the L2 and L-infinity norm.

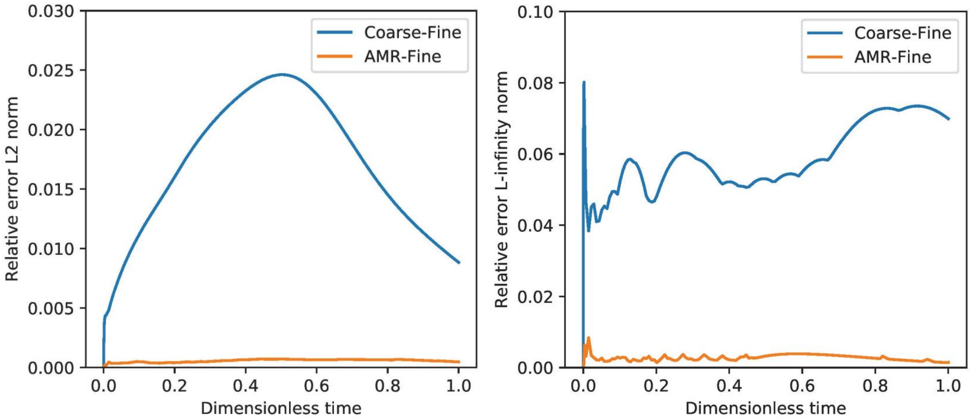


Figure 3.3: L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the homogeneous model with unstructured gridding from figures 3.1 to 3.2.

The graph below shows the percentage of grid cells used in the simulation using the AMR grid, relative to the total number of cells in the fine-scale model. As can be seen, the amount of cells is considerably reduced, ranging from 8 to 60%. The trend shows an overall increase as the front propagates, and a decrease when the cold front has reached the producing well, which results in coarsening at locations where no more thermal variations are detected.

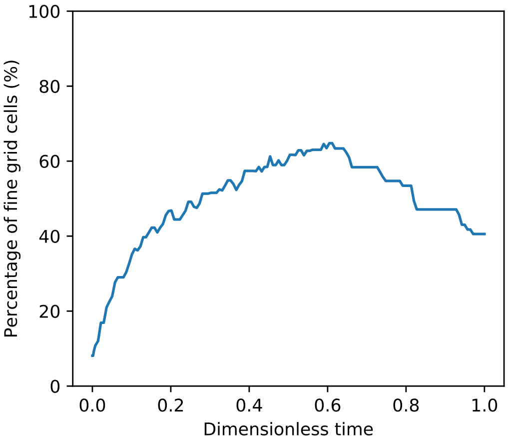


Figure 3.4: Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the homogeneous model with unstructured gridding from figures 3.1 to 3.2.

This considerably improves the performance of simulation since the AMR approach is much more favourable in terms of efficient use of computational resources. The cold water front is well-captured throughout the simulation and the results are close to the fine-scale solution.

3.1.2. Sugar-cube shale model

Shales are often neglected in conventional reservoir simulation as the convective flow is never acquired in shales due to low permeability. However, for geothermal applications, they represent an important source of heat which recharges the flowing cold water (thermal recharge). It is therefore important to model the shale bodies which are in close proximity to the sand channels. However, modelling of the shales significantly increases computational time since shales often occupy a significant amount of computational grid. Here, we test an application of our AMR approach to a sugar-cube model where cubes represents shale bodies and space between them fluvial channels.

We use a simple 2D setup to illustrate the effectiveness of the AMR solution. In total, a 5 by 6 shale block configuration is used. Shale blocks have a permeability of 10^{-2} mD while the sand bodies have a permeability of 10^{3} mD. The injector and producer are placed at opposite corners of the reservoir as shown in the figure 3.5. Level 0 consists of 4588 fine cells. Level 1 is constructed differently from conventional AMR approach with the sand channels kept at fine level, and only the shale blocks are coarsened by a ratio of 100 (10 × 10). The coarse grid, level 1, contains 1618 cells. Both levels are shown in figure 3.5 below.

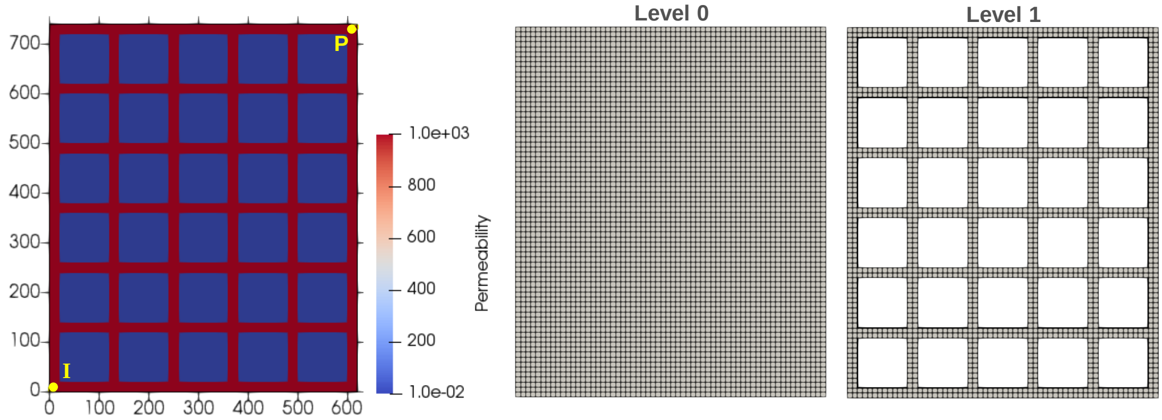


Figure 3.5: Hierarchical grid of the model with shale blocks. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

Figure 3.6 below depicts the temperature solution at three different times throughout the simulation: t_D = 0.1, 0.3 and 0.7. Similarly to the previous example, the solution is shown for (a) the fine grid, (b) the AMR grid and (c) the coarse grid.

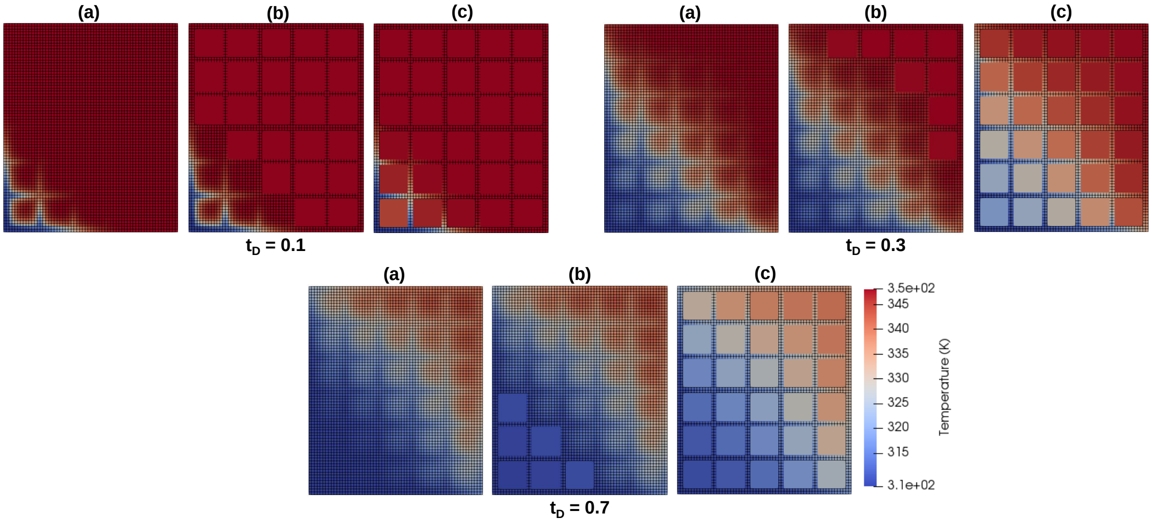


Figure 3.6: Temperature solution of the shale blocks model at three different times: t_D = 0.1, 0.3 and 0.7. (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

As can be seen on the AMR figure (b), the grid refines as soon as the cold front arrives at proximity to a shale body. The cold front is accurately represented on the AMR grid and there are no differences compared to the fine grid. On the coarse grid however, the cold front propagates further than for the fine and AMR model, which is clearly visible at the late time recording $t_D = 0.7$ in the figure above. When the cold front passes part of the shales blocks and these have cooled down, coarsening occurs as observed at $t_D = 0.7$.

An error analysis was performed to investigate the relative error of the AMR and coarse model where the fine-scale solution is taken as reference. Figure 3.7 below depicts the error distribution through time of both the AMR and coarse model relative to the fine model.

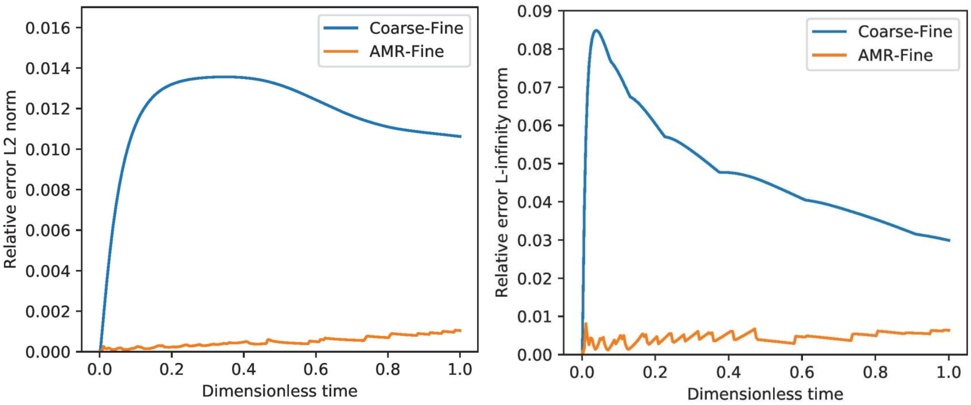


Figure 3.7: L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the shale blocks model from figures 3.5 to 3.6.

Both the L2 norm and the L-infinity norm are plotted. As can be seen, the error of the coarse model is significantly larger than for the AMR model, where the error is close to zero. The percentage of cells used in the AMR grid relative to the amount of cells used in the fine grid is shown in figure 3.8 below. As can be seen, the percentage of cells ranges from roughly 35% to around 90% halfway through the simulation, when the cold front reaches the producing well, and then lowers to 65% when shale blocks proximal to the injector wells have cooled down to injection temperature, and hence coarsening occurs.

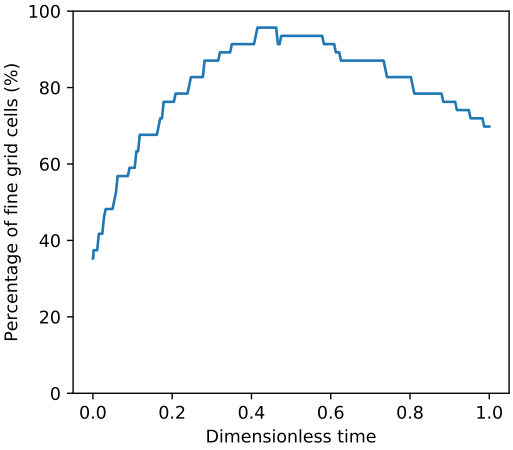


Figure 3.8: Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the shale blocks model from figures 3.5 to 3.6.

As observed, the computational time and effort was considerably reduced throughout the simulation, and the AMR solution outcome shows a very accurate representation of the fine-scale model.

3.1.3. Fluvial heterogeneous model

Our AMR framework was tested for a heterogeneous reservoir with a low net-to-gross ratio (N/G = 35%). The permeability field ranges from 0.1 to 3400 mD with a significant amount of shale regions present. The hierarchical grid for this example is a structured grid and it comprises two levels. The finest grid, level 0, consists of 2400 grid cells with 40 cells in the x-direction and 60 cells in the y-direction. Level 1 was reduced to 150 mesh elements, where aggregation was done using 4x4 fine mesh elements, resulting in 10 grid cells in the x-direction and 15 grid cells in the y-direction. The permeability field along with the hierarchical grid for this example is shown in figure 3.9 below. The location of the injector (I) and producer (P) are depicted in yellow on the permeability distribution figure below.

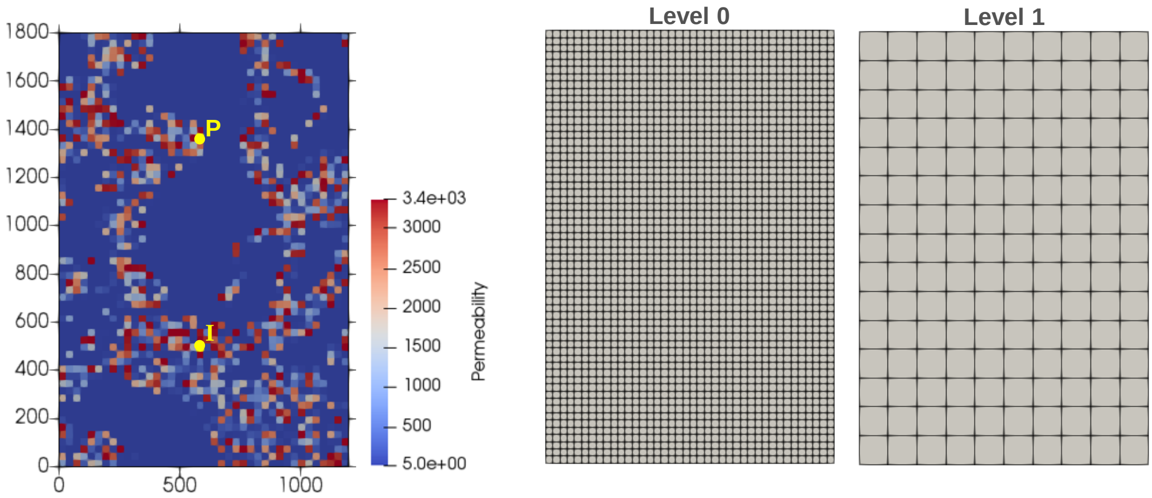


Figure 3.9: Hierarchical grid of the heterogeneous model with low net-to-gross ratio. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

The simulation was conducted until cold water breakthrough reached the producing well. Figure 3.10 below illustrates the temperature solution at different times throughout the simulation. For each time shown, figure (a) represents the fine-scale solution, figure (b) is the AMR solution, and figure (c) is the coarse-scale solution. The grid is kept at its finest level at well locations.

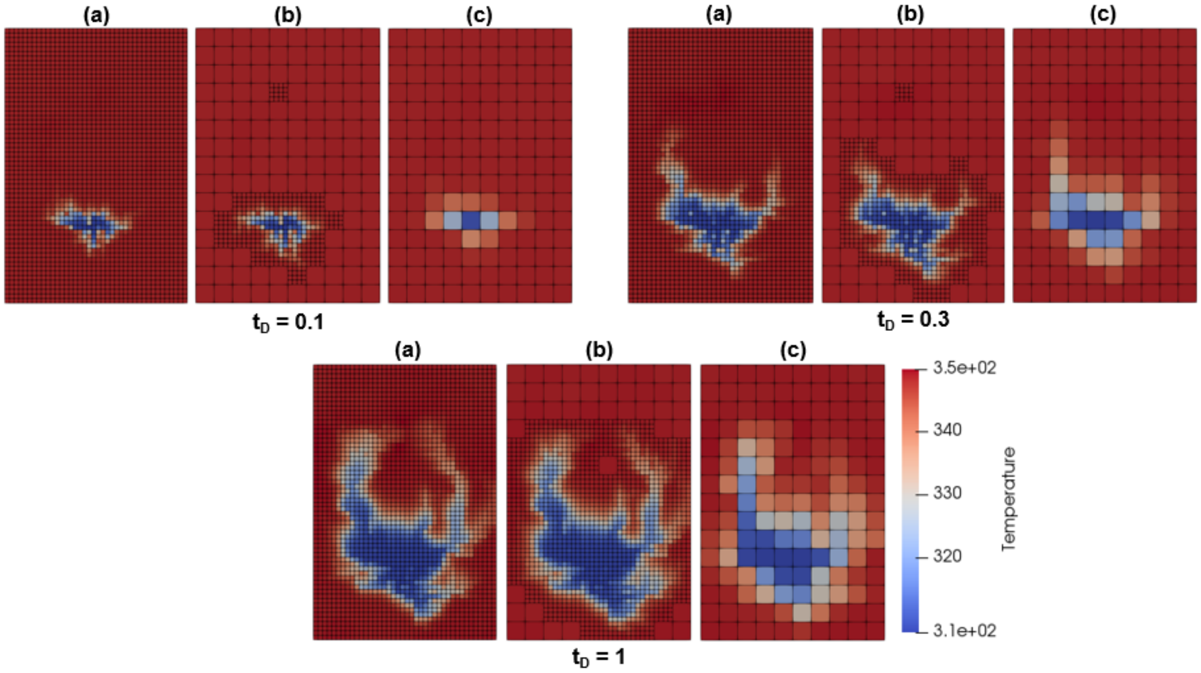


Figure 3.10: Temperature solution of the heterogeneous model with low net-to-gross ratio at three different times: at t_D = 0.1, 0.3 and 1. (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

As can be seen, the AMR mesh exhibits a significant improvement in temperature solution compared to the solution on the coarse grid. Refinement is mainly focused at the front and slightly beyond the front, while areas where insignificant changes occur remain coarse. Important details, such as fingering effects at the cold water front, which are neglected on the coarse grid, are clearly visible in both fine and AMR solutions, which results in a more accurate representation of this physical phenomenon.

The relative error throughout the simulation run was recorded, where the fine model is taken as reference solution, for comparison between the coarse and AMR model. Figure 3.11 below shows the L2 norm and the L-infinity norm error in time.

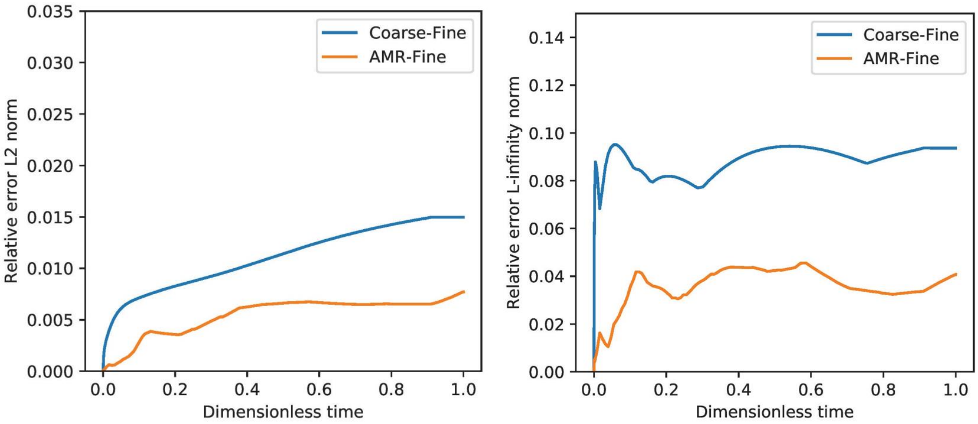


Figure 3.11: L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the heterogeneous model with low net-to-gross ratio from figures 3.9 to 3.10.

As can be seen, the marked improvement is also recorded in the error analysis, where the error between the coarse and fine model is notably larger than the error between the AMR and fine model. The L2 norm remains relatively constant for the AMR solution whereas it increases slightly in time for the coarse solution.

The graph in figure 3.12 below shows the amount of grid-cells used in the simulation where AMR was implemented, compared to the fine-scale model, expressed in percentage.

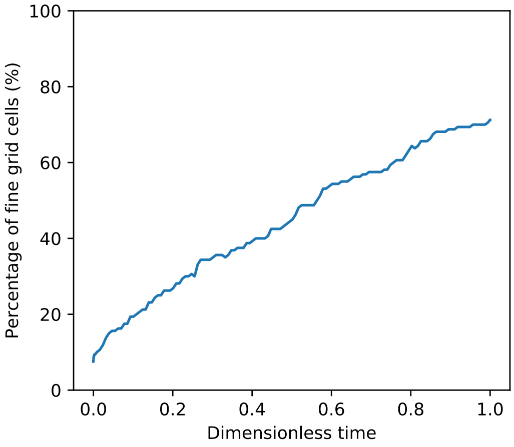


Figure 3.12: Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the heterogeneous model with low net-to-gross ratio from figures 3.9 to 3.10.

As can be seen, the percentage of fine grid cells in the AMR grid ranges from 8 to 70% throughout the simulation. This represents a significant improvement in computational effort and time, while still capturing important features.

3.1.4. SPE10 reservoir

In this example, four highly heterogeneous fluvial layers of the SPE 10 model from layer 79-th to 82-nd were used (M. Christie & Blunt, 2001). This 3D model consists of a channeled system with high permeability channels and low permeability shales around the channels. The permeability ranges from 0.0015 to 20000 mD as shown in figure 3.13 for layer 79 using the logarithm of the permeability to emphasize the contrast between geological features. Level 0 consists of 52800 cells with 60, 220 and 4 cells in the x-, y- and z-direction respectively. Level 1 has been reduced to 825 cells with an aggregation of 4 fine grid cells in all 3 directions. Both levels are shown in figure 3.13 below.

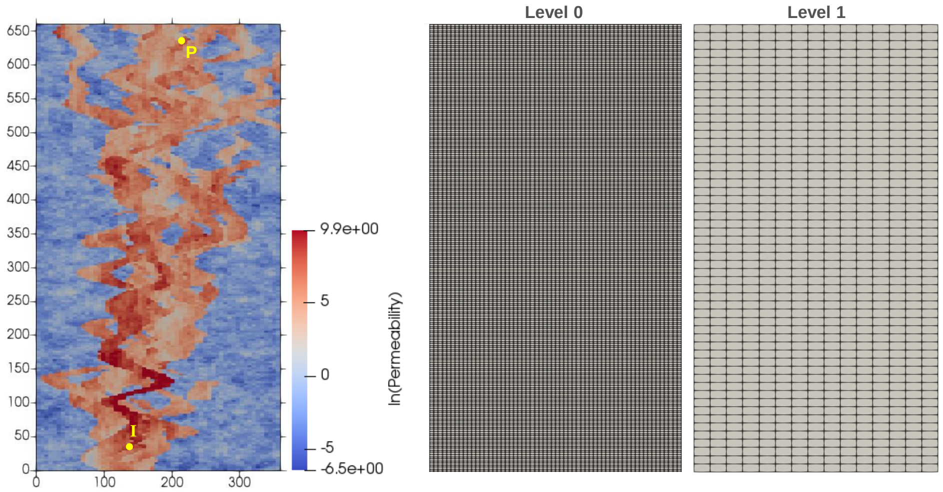


Figure 3.13: 2D overview of the hierarchical grid of the SPE10 reservoir (layers 79 to 82). Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

The simulation was conducted for a period of 500 days. Figure 3.14 below illustrates the temperature solution at three different times with (a) at the fine scale, (b) on the AMR grid and (c) at the coarse scale.

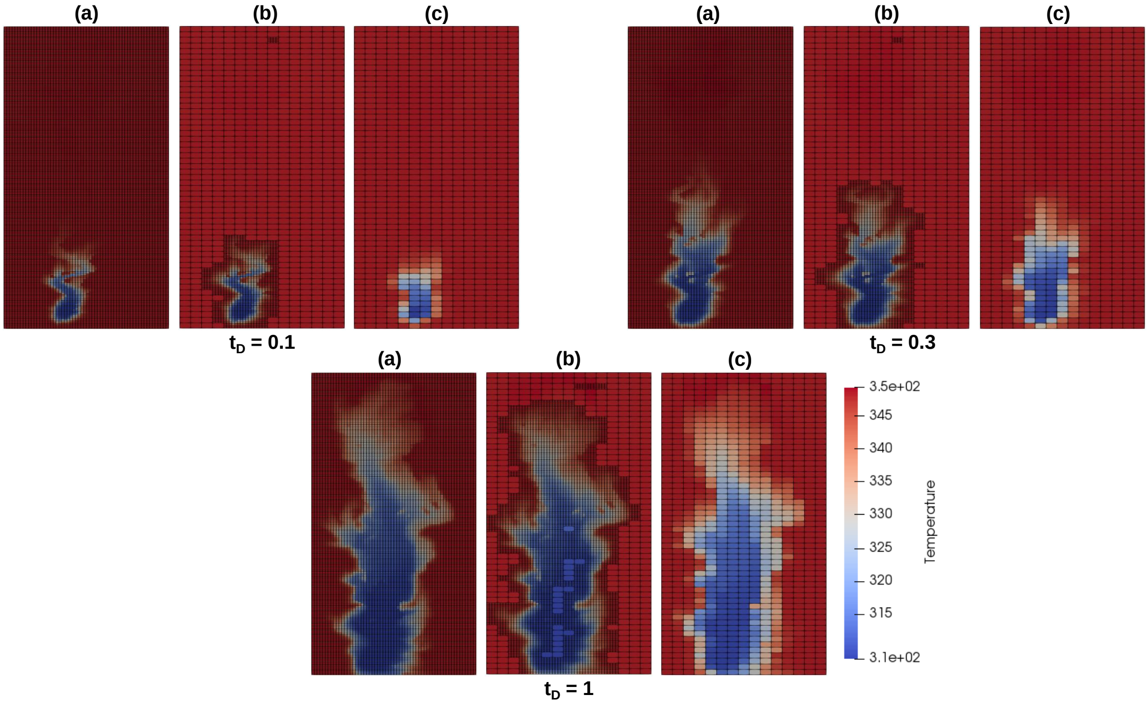


Figure 3.14: Temperature solution of the SPE10 reservoir at three different times. (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

As can be seen, the AMR solution shows significant improvements in the solution as compared to the coarse-scale solution. The coarse-scale solution neglects important fine-scale features along the cold front. It can be observed that mesh refinement occurs along the cold front, where significant changes are observed, but the mesh remains coarse at locations where no major changes occur. This improves the efficiency of computations.

An error analysis was also conducted in order to record the performance of the AMR framework relative to the fine-scale solution. Figure 3.15 below depicts the L2 and L-infinity norm of the error throughout the simulation.

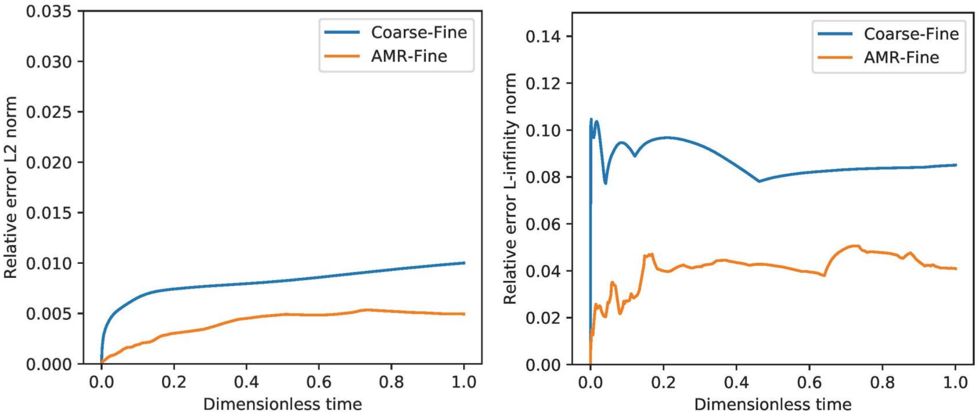


Figure 3.15: L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the SPE10 reservoir model from figures 3.13 to 3.14.

The graphs show that the error resulting from the AMR model is significantly less than the error resulting from the coarse-scale model, for both the L2 and L-infinity norm. The L2 norm relative error reaches a maximum of 0.5% for the AMR model and 1% for the coarse-scale solution. The L-infinity norm relative error is around 0.04 for the AMR model, whereas it is around 0.09 for the coarse model. The error is therefore halved when using the AMR model.

The performance of the AMR framework can also be illustrated through determining the percentage of grid cells of the fine-scale model used during the simulation run on the AMR grid. The graph in figure 3.16 below depicts this percentage throughout the simulation.

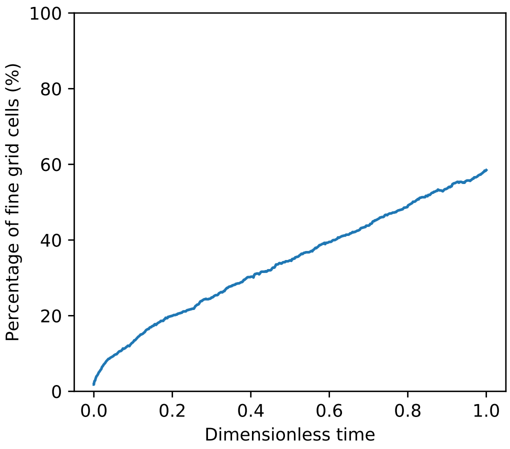


Figure 3.16: Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the SPE10 reservoir model from figures 3.13 to 3.14.

As shown in the figure above, the number of grid cells present in the AMR grid ranges from 2 to 55% of the fine-scale model. This represents a considerable improvement in efficient use of computational resources.

3.2. Reactive transport

Carbonate reservoirs host a major part of the world's hydrocarbon reserves. But besides hydrocarbon reserves, the ongoing energy transition has resulted in an increase interest in geothermal systems where many are hosted by carbonate rocks. These reservoirs can have heavily fractured and karstified intervals, resulting in unforeseen hazards during drilling. Furthermore, naturally fractured carbonate reservoirs contain a large uncertainty in flow response due to the poor ability to predict the spatial distribution of discontinuity networks at reservoir-scale.

Another important process related to dissolution is well acidization used to increase the production. This process involves the dissolution of reservoir rock to stimulate flow towards the wells. These chemical reactions are localized and form important features for accurate representation of the flow response. Furthermore, reaction rates which occur during dissolution are high, resulting in a sharp front in the flow response. Moreover, during dissolution, formation and development of an unstable dissolution front with multiple wormholes can occur and its modeling is quite sensitive to the resolution (Shaik et al., 2018). In near-well acidization processes, the regime which forms a single dominating wormhole is the most preferable. It is therefore important to accurately predict this unstable dissolution while keeping the computational time reasonable. AMR is therefore a good solution to model these reservoirs and chemical processes to solution requirements.

In the flow example analyzed in this study, dissolution involves the following simple reaction where carbonate is dissolved:

$$\mathrm{CaCO_{3(s)}} \longrightarrow \mathrm{Ca^{2+}} + \mathrm{CO_3^{2-}} \tag{3.2}$$

The model we use simulates the phenomenon of unstable wormhole formation triggered by small perturbations in permeability. On one side of the reservoir, an injector well is placed which is perforated throughout the whole thickness. On the other side, the producer well is placed, also spanning the entire thickness of the reservoir. The model described in this example has dimensions of 100 by 100 meters. A constant permeability of 1 mD is used with 5% of random noise. The left illustration in figure 3.17 shows the well locations, along with the permeability of the reservoir. The hierarchical grid consists of two levels, where level 0 is an unstructured grid containing 2194 triangular cells. Cell aggregation was conducted to construct level 1, where the x- and y- axes were divided in 10 equidistant intervals, resulting in a grid with only 100 cells. Level 0 and 1 are shown in figure 3.17.

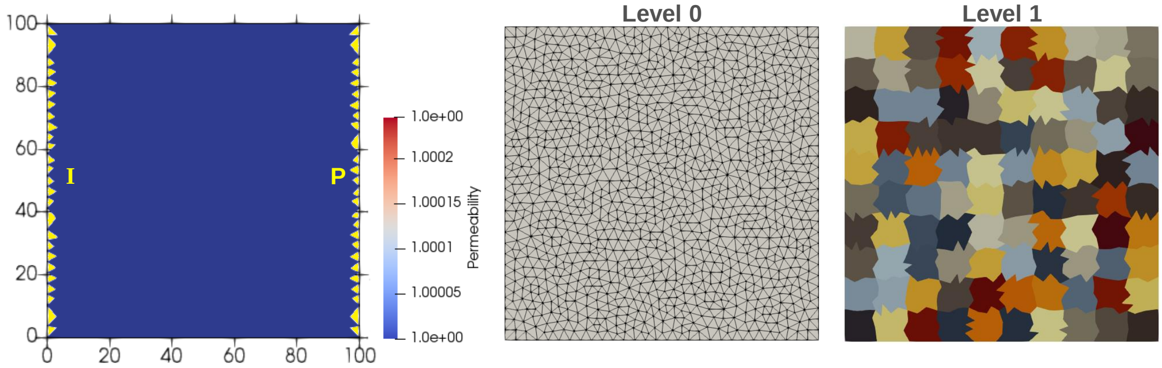


Figure 3.17: Hierarchical grid of the wormhole model. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

The AMR simulation was started at the coarse level, while keeping the well cells at the finest level throughout the entire simulation run. For this application, the adaptivity criterion is based on the solid composition, $x[CaCO_3]$. The adaptivity criteria used in this example are defined as follows:

$$\begin{cases} \text{if } \Delta \mathrm{x[CaCO_3]}_{ij} > 0.02, & \text{mark cells i and j for refinement,} \\ \text{if } \Delta \mathrm{x[CaCO_3]}_{l} < 0.01, \quad \forall l \in I, & \text{mark cells } \{\forall i \in I\} \text{ for coarsening,} \end{cases} \tag{3.3}$$

where $\Delta x[CaCO_3]$ is the difference in composition of the calcium carbonate $CaCO_3$ component.

The simulation was recorded at three different times: 0.13, 0.25 and at the final time (expressed in dimensionless time). Figure 3.18 depicts the solid composition - the composition of $CaCO_3$ - in time, where figure

(a) is the fine-scale solution, figure (b) is the AMR solution, figure (c) is the coarse-scale solution, and figure (d) represents the node distribution of the AMR grid.

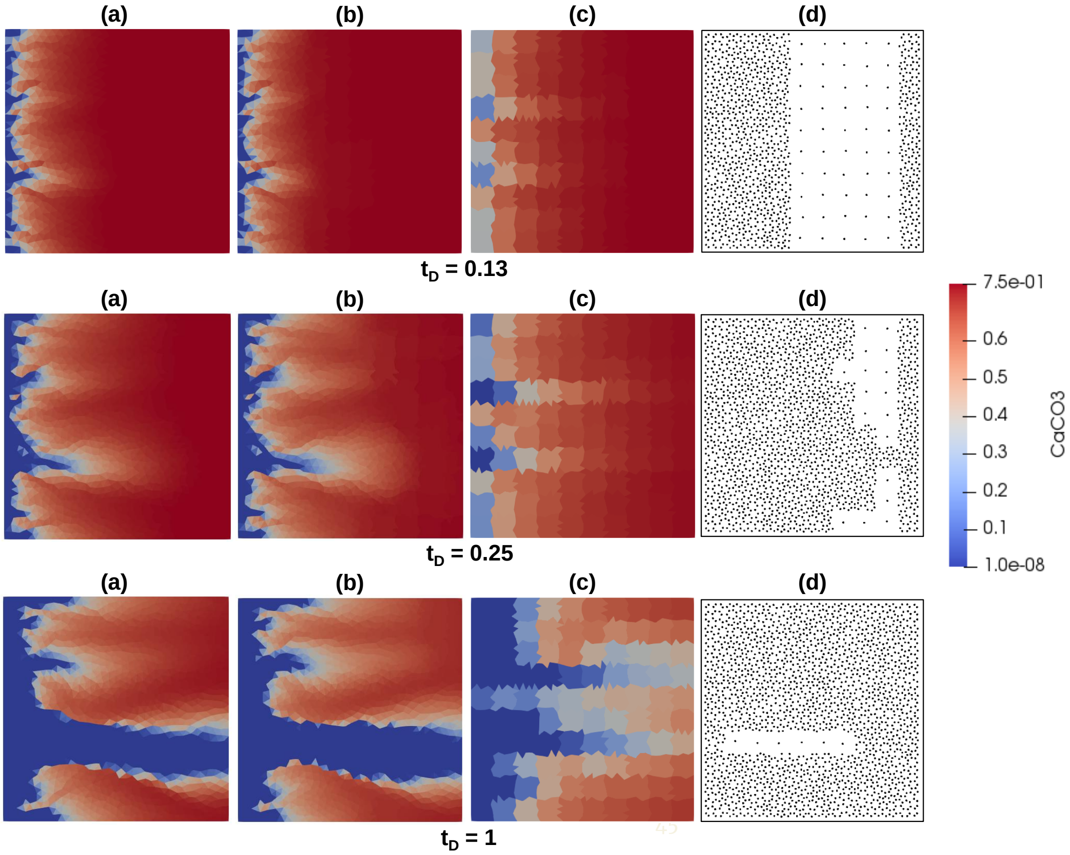


Figure 3.18: Solid composition solution of the wormhole model at three different times. (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

As can be seen, the AMR solution is considerably more accurate than the coarse-scale solution. The far-propagating wormhole (at $t_D = 1$), which is present in both the fine-scale solution and the AMR solution, is not well represented on the coarse-scale solution, where two extensive wormholes are present. The AMR solution however shows a very good representation of the fine-scale solution throughout time. The most extensive wormhole exhibits slight differences in thickness and some minor variations are observed at the other smaller wormholes. The node distribution follows the front, which is in this example quite dispersed, resulting in refinement spanning a wide area, especially at the last time step. However, considerable computational resources are saved at the early stage of the simulation.

To quantify the differences between fine-scale solution, AMR solution and coarse-scale solution, an error analysis was conducted. Here again, both the L2 norm and the L-infinity norm were computed for the AMR and coarse model, relative to the fine-scale solution. The graphs in figure 3.19 below depict the outcome. As can be seen, the AMR error is once more significantly less than the coarse model, for both norms. For the L2 norm, the coarse-fine relative error is three times greater than the AMR-fine error at the final time step. The L-infinity norm of the coarse-fine error starts low at the first time step, where no extensive propagation is observed and where the model is close to the initial conditions, but then rapidly increases to 0.8 and remains more or less constant throughout. The L-infinity norm of the AMR-fine error seems to increase in time. This is due to the propagation of initially small errors in the solution. Note however that the relatively big error for both the AMR and coarse-scale model are not representative for this example and are related to another type of instability in the solution.

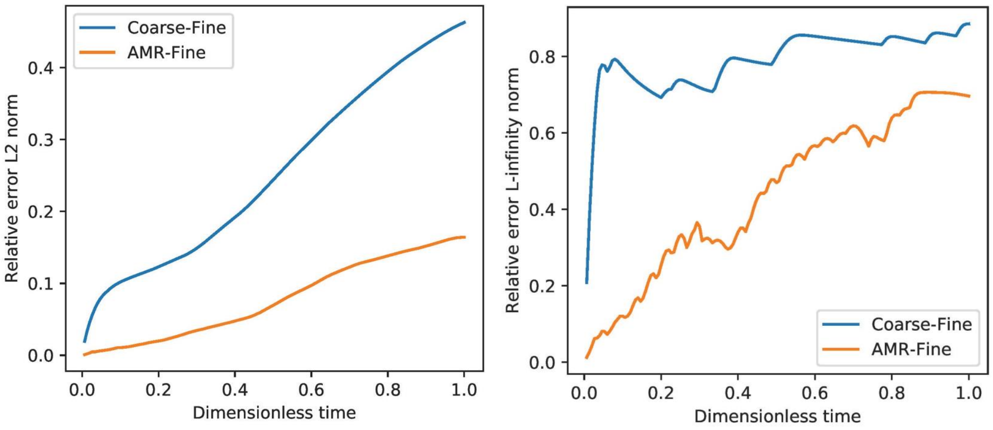


Figure 3.19: L2 norm and L-infinity norm of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the wormhole model from figures 3.17 to 3.18.

Similarly to the previous example, we have analyzed the total number of cells used in the AMR model, relative to the total amount of cells contained in the fine-scale grid. The graph in figure 3.20 below shows this quantity expressed in percentage.

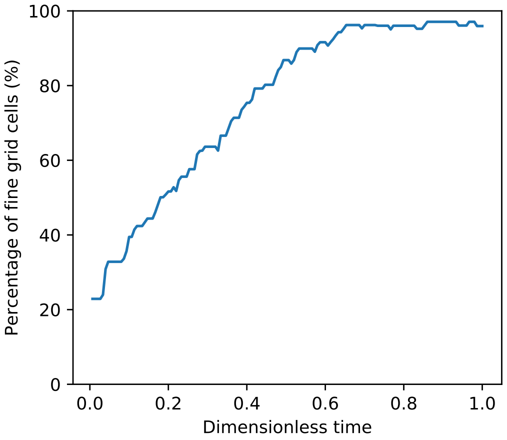


Figure 3.20: Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the wormhole model from figures 3.17 to 3.18.

As can be seen, the amount of cells used during the AMR simulation is overall less than the amount of cells present in the fine-scale model. Initially, the amount of cells starts at 20%, which represents the use of the coarsest level, with both left and right boundaries kept at the finest level. It then increases, fairly steeply, to around 95% due to the high injection velocity which corresponds to the dominating wormhole regime. Around the end of the simulation, the amount of cells starts to decrease, which indicates coarsening at some locations. Although almost 100% of the cells is used at two thirds of the simulation, which is computationally expensive, considerable resources are saved in the beginning. Moreover, this problem is sensitive to the resolution which requires refinement at many locations in order to accurately capture the wormhole propagation.

4

Concluding remarks

This study aimed at developing an Adaptive Mesh Refinement (AMR) technique in *Delft Advanced Research Terra Simulator* for general-purpose reservoir simulation. The developed AMR framework consists of a multi-level hierarchical grid, where levels are constructed through a mesh partitioning of the fine-scale model - the static geological model - which is represented by a list of properties (e.g. volume and porosity). The framework consists of the construction of the coarse levels through cell aggregation of the next consecutive fine level at the pre-processing stage. The method used to aggregate fine cells includes the grouping of subdomains whose centroids are found within a predefined 3D domain. In this study, domains are grouped by the partitioning of the x-, y- and z- axes into equidistant intervals.

The aggregation of the subdomains to form a coarser level is stored as a list of indices for next stages, which consists of the indices of the fine cells comprised in its coarse control volume for each coarse cell. Next, in order to solve the relevant governing equations, the flow must be computed at each interface present in the mesh. We therefore generate a list - called a connectivity list - describing all neighbouring cells within each level and between levels. The fine-scale transmissibility is then computed using the permeability field. Hereafter, a flow-based upscaling is applied in order to acquire the transmissibility of coarser levels, and the inter-level transmissibility. Each control volume has defined parameters which are relevant for solving the system (volume, porosity, depth etc).

Once the hierarchy of levels is complete, the simulation can be started. The first time step is performed on the coarsest level, while keeping the well blocks at the finest level. Adaptivity check is performed at every time step, using criteria specific to the application. Once the regions for coarsening and refinement are defined, the solution is prolongated to the finest meshing level, and subsequently restricted from fine to the adaptive simulation grid. A new connection list and grid properties are constructed for the new coarsened schema. Once it is completed, the simulation runs for the next time step using the constructed simulation model.

The accuracy of the method was demonstrated for several problems using two different applications: geothermal energy extraction and reactive flow and transport. For geothermal applications, four models were tested, including a homogeneous model with unstructured gridding, a synthetic sugar-cube-like model with high permeability channels surrounded by shale blocks, a heterogeneous fluvial system model with low net-to-gross ratio, and finally, the framework was tested on several layers of a highly heterogeneous SPE10 reservoir. High levels of solution accuracy relative to the reference fine-scale results are observed for all cases. An error analysis was conducted to record the differences between the AMR and the coarse solution relative to the reference fine-scale solution. The error resulting from the AMR model is significantly lower than for the coarse model, for all tested problems. The overall percentage of grid cells used in the AMR model relative to the fine-scale model is considerably decreased for most problems.

For reactive flow and transport, the dissolution of calcium carbonate rock was analyzed for a small-scale reservoir where small perturbations in permeability induce the formation of an unstable dissolution front with wormholes. Here also, the AMR solution shows high accuracy relative to the reference fine-scale solution. The error is approximately a third of the coarse-scale solution error. The quantity of grid cells used throughout the simulation run does unfavourably reach close to 100% of the fine-scale grid due to the small characteristic domain of the model. We expect this will be improved for a larger, more realistic domain.

To conclude, the developed AMR method shows high levels of accuracy for both homogeneous and heterogeneous models, and can be used for geothermal applications as well as for reactive flow and transport with dissolution. The number of cells in the AMR simulation, relative to the total number of cells of the finest level, is considerably reduced, which is very favourable in terms of efficient use of computational resources. The framework is applicable to two- and three-dimensional models and for unstructured as well as structured meshes. The applicability of the method to unstructured grids provides an effective means for solving complex geological systems.

For future work, this technique can be extended to discrete fracture models, where local refinement can significantly improve the accuracy in the representation of fractures. Moreover, the AMR framework should be tested on other applications where localization occurs, and where higher resolution is required, e.g. in-depth water divergence (Trujillo et al., 2018). Another possible improvement in the AMR solution is to use an interpolation method for the prolongation operation. This can potentially facilitate convergence of the solution and result in a better representation of the solution outcome.

References

Bahrainian, S. S., & Dezfuli, A. D. (2014). A geometry-based adaptive unstructured grid generation algorithm for complex geological media. *Computers & geosciences*, *68*, 31–37.

Branets, L. V., Ghai, S. S., Lyons, S. L., & Wu, X.-H. (2009). Challenges and technologies in reservoir modeling. *Communications in Computational Physics*, *6*(1), 1.

Christensen, J., Darche, G., Dechelette, B., Ma, H., Sammon, P., et al. (2004). Applications of dynamic gridding to thermal simulations. In *SPE international thermal operations and heavy oil symposium and western regional meeting.*

Christie, M., & Blunt, M. (2001). Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Evaluation and Engineering, 4*(4), 308-316.

Christie, M. A. (1996). Upscaling for reservoir simulation. *Journal of petroleum technology, 48*(11), 1–004.

Cusini, M., van Kruijsdijk, C., & Hajibeygi, H. (2016). Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media. *Journal of Computational Physics, 314*, 60–79.

DARTS. (2019). Delft Advanced Research Terra Simulator. In *https://darts-web.github.io/darts-web/.*

Durlofsky, L. J. (2005). Upscaling and gridding of fine scale geological models for flow simulation. In *8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy* (Vol. 2024, pp. 1–59).

Gong, B., Karimi-Fard, M., & Durlofsky, L. J. (2008). Upscaling discrete fracture characterizations to dual-porosity, dual-permeability models for efficient simulation of flow with strong gravitational effects. *SPE Journal*, *13*(01), 58–67.

Jenny, P., Lee, S., & Tchelepi, H. A. (2003). Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics, 187*(1), 47–67.

Karimi-Fard, M., & Durlofsky, L. J. (2012). Accurate resolution of near-well effects in upscaled models using flow-based unstructured local grid refinement. *SPE Journal, 17*(04), 1–084.

Karimi-Fard, M., & Durlofsky, L. J. (2014). Unstructured adaptive mesh refinement for flow in heterogeneous porous media. In *ECMOR XIV-14th European conference on the mathematics of oil recovery.*

Karimi-Fard, M., Gong, B., & Durlofsky, L. J. (2006). Generation of coarse-scale continuum flow models from detailed fracture characterizations. *Water resources research, 42*(10).

Khait, M., & Voskov, D. V. (2017). Operator-based linearization for general purpose reservoir simulation. *Journal of Petroleum Science and Engineering, 157*, 990–998.

Lim, K.-T., Schiozer, D., & Aziz, K. (1995). A new approach for residual and jacobian arrays construction in reservoir simulators. *SPE Computer Applications, 7*(04), 93–96.

Pau, G. S. H., Bell, J. B., Almgren, A. S., Fagnan, K. M., & Lijewski, M. J. (2012). An adaptive mesh refinement algorithm for compressible two-phase flow in porous media. *Computational Geosciences, 16*(3), 577–592.

Sammon, P. H., et al. (2003). Dynamic grid refinement and amalgamation for compositional simulation. In *SPE reservoir simulation symposium.*

Shaik, A., Tomin, P., & Voskov, D. (2018). Modeling of near-well matrix acidization. In *43rd workshop on geothermal reservoir engineering.*

Shetty, S., Voskov, D. V., & Bruhn, D. (2017). *Numerical strategy for uncertainty quantification in low enthalpy geothermal projects* (Unpublished doctoral dissertation). M. Sc. thesis. Department of Applied Earth Science. Delft University of Technology.

Trangenstein, J. A. (2002). Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Advances in Water Resources*, *25*(8-12), 1175–1213.

Trujillo, R., Voskov, D., & Leeuwenburgh, O. (2018). Simulation of In-depth Water Diversion Using Sodium Silicate. In *43rd workshop on geothermal reservoir engineering.*

Van Batenburg, D. W., De Zwart, A., Boerrigter, P. M., Bosch, M., & Vink, J. C. (2011). Application of dynamic gridding techniques to IOR/EOR processes. In *IOR 2011-16th European Symposium on Improved Oil Recovery.*

Voskov, D. V. (2017). Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. *Journal of Computational Physics*, *337*, 275–288.

Wang, Y., Hajibeygi, H., & Tchelepi, H. A. (2014). Algebraic multiscale solver for flow in heterogeneous porous media. *Journal of Computational Physics*, *259*, 284–303.

Willems, C. (2017). *Doublet deployment strategies for geothermal hot sedimentary aquifer exploitation: application to the Lower Cretaceous Nieuwerkerk Formation in the West Netherlands Basin* (Unpublished doctoral dissertation). Delft University of Technology.