

Accurate geothermal and chemical dissolution simulation using adaptive mesh refinement on generic unstructured grids

de Hoop, Stephan; Jones, Elodie ; Voskov, Denis

DOI

[10.1016/j.advwatres.2021.103977](https://doi.org/10.1016/j.advwatres.2021.103977)

Publication date

2021

Document Version

Final published version

Published in

Advances in Water Resources

Citation (APA)

de Hoop, S., Jones, E., & Voskov, D. (2021). Accurate geothermal and chemical dissolution simulation using adaptive mesh refinement on generic unstructured grids. *Advances in Water Resources*, 154, 1-17. Article 103977. <https://doi.org/10.1016/j.advwatres.2021.103977>

Important note

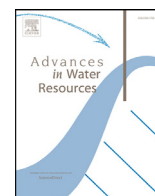
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



Accurate geothermal and chemical dissolution simulation using adaptive mesh refinement on generic unstructured grids

Stephan de Hoop^a, Elodie Jones^a, Denis Voskov^{a,b,*}

^a Delft University of Technology, Department of Geo-sciences & Engineering, Stevinweg 1, 2628 CN, Delft, the Netherlands

^b Stanford University, Department of Energy Resources Engineering, 367 Panama street, Stanford, CA 94305, United States

ARTICLE INFO

Keywords:

Geothermal simulation
Adaptive Mesh Refinement
Unstructured grid
DARTS

ABSTRACT

A coupled description of flow and thermal-reactive transport is spanning a wide range of scales in space and time, which often introduces a significant complexity for the modelling of such processes. Subsurface reservoir heterogeneity with complex multi-scale features increases the modelling complexity even further. Traditional multiscale 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 thermal-compositional flow. The use of an Adaptive Mesh Refinement enables the grid to adapt dynamically during the simulation, which facilitates the efficient use of computational resources. This is especially important in applications with thermal flow and transport where the region requires high-resolution calculations as often localized in space. In this work, the aim is to develop an Adaptive Mesh Refinement framework for geothermal reservoir simulation. The approach uses a multi-level connection list and can be applied to fully unstructured grids. The adaptivity of the grid in the developed framework is based on a hierarchical connectivity list. First, the fine-scale model is constructed, which accurately approximates all reservoir heterogeneity. Next, a global flow-based upscaling is applied, where an unstructured partitioning of the original grid is created. Once the full hierarchy of levels is constructed, the simulation is started at the coarsest grid. Grid space refinement criteria is based on the local changes and can be adjusted for specific models and governing physics. The multi-level connectivity lists are redefined at each timestep and used as an input for the next. The developed Adaptive Mesh Refinement framework was implemented in Delft Advanced Research Terra Simulator which uses the Operator-Based Linearization technique. The performance of the proposed approach is illustrated for several challenging geothermal applications of practical interest.

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 (Christie, 1996).

An issue of considerable importance is the risk and uncertainty assessment of reservoir performance. The uncertainty can be gauged by simulating an ensemble of different geological realizations (Chen et al., 2015). 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 (Durlafsky, 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 Multiscale techniques (Jenny et al., 2003; Wang et al., 2014), developed to solve the elliptic (or parabolic) pressure equation in sequentially coupled simulations, mainly focus on the pressure solu-

* Corresponding author.

E-mail address: d.v.voskov@tudelft.nl (D. Voskov).

<https://doi.org/10.1016/j.advwatres.2021.103977>

Received 3 February 2021; Received in revised form 27 May 2021; Accepted 3 June 2021

Available online 10 June 2021

0309-1708/© 2021 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

tion 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 and Durlofsky, 2014).

In today's literature, several researchers have developed and proposed AMR procedures to capture the local nature of transport processes. Bahrainian and Dezfouli (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), Discrete Fracture Networks (Berrone et al., 2019), and many more applications.

In this work, the aim was to develop a dynamic AMR scheme using an unstructured multi-level gridding framework, for geothermal simulation in complex reservoirs. The focus lied particularly on thermal-reactive flow and transport formulation which are required for a wide range of subsurface applications relevant to the energy transition including geothermal. Notice that heterogeneity plays a very important role in geothermal applications (Shetty et al., 2018; Babaei and Nick, 2019). The geothermal doublet lifetime and heat recovery rate usually vary a lot with both reservoir parameters and operational management where uncertainties due to heterogeneity are dominating (Willems and M. Nick, 2019). Heterogeneity in flow path and shale facies play an important role in water heat recharge which directly affects doublet performance at low net-to-gross ratio (Crooijmans et al., 2016). Besides, complex heat extraction process and corresponding chemical interactions can also amplify the effect of heterogeneity (Cui et al., 2016; Kala and Voskov, 2020).

As a starting point of our framework, a fine-scale geological model has to be constructed accurately approximating all reservoir heterogeneity. In reservoir simulation, this model is often represented by an array 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 and Durlofsky, 2014). This partitioning provides coarser levels of the original model which is also described by an array 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) (Kala and Voskov, 2020; Wang et al., 2020).

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 (Khaib and Voskov, 2018b). When chemical reactions occur in the system, an additional term describing n_k kinetic reactions is added to the mass conservation equation (Kala and Voskov, 2020). These governing relations are described as:

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

$$\begin{aligned} \frac{\partial}{\partial t} \left(\phi \sum_{p=1}^{n_p} \rho_p s_p U_p + (1 - \phi) U_r \right) + \text{div} \sum_{p=1}^{n_p} h_p \rho_p \bar{u}_p + \text{div}(\kappa \nabla T) \\ + \sum_{p=1}^{n_p} h_p \rho_p \bar{q}_p = 0, \end{aligned} \quad (2)$$

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 Eq. (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 ξ and/or physical state ω :

$\phi(\xi, \omega)$	porosity,
$x_{cp}(\omega)$	the mole fraction of component c in phase p ,
$s_p(\omega)$	phase saturation,
$\rho_p(\omega)$	phase molar density,
$\bar{u}_p(\xi, \omega)$	phase velocity,
$\bar{q}_p(\xi, \omega, \mathbf{u})$	source of phase p ,
$U_p(\xi)$	phase internal energy,
$U_r(\xi)$	rock internal energy,
$h_p(\xi)$	phase enthalpy,
$\kappa(\xi, \omega)$	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 ξ only. Phase flow velocity $\bar{u}_p$ is assumed to follow Darcy's law, expressed as:

$$\bar{u}_p = - \left(\frac{k_{rp}}{\mu_p} (\nabla P_p - \bar{\gamma}_p \nabla D) \right), \quad p = 1, \dots, n_p, \quad (3)$$

where

$\mathbf{K}(\xi)$	permeability tensor,
$k_{rp}(\omega)$	relative permeability of phase p ,
$\mu_p(\omega)$	phase viscosity,
$P_p(\omega)$	pressure in phase p ,
$\bar{\gamma}_p(\omega)$	gravity vector,
$D(\xi)$	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 Eqs. (1) and 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 $\bar{u}_p$ are substituted by the Darcy relation (3):

$$V \left[\left(\phi \sum_{p=1}^{n_p} x_{cp} \rho_p s_p \right)^{n+1} - \left(\phi \sum_{p=1}^{n_p} x_{cp} \rho_p s_p \right)^n \right] - \Delta t \sum_l \left(\sum_{p=1}^{n_p} x_{cp} \rho_p^l \Gamma_p^l \Delta \psi^l \right) + \Delta t \sum_{p=1}^{n_p} \rho_p x_{cp} q_p = V \Delta t \sum_k v_{ck} r_k, \quad c = 1, \dots, n_c, \quad (4)$$

$$V \left[\left(\phi \sum_{p=1}^{n_p} \rho_p s_p U_p + (1 - \phi) U_r \right)^{n+1} - \left(\phi \sum_{p=1}^{n_p} \rho_p s_p U_p + (1 - \phi) U_r \right)^n \right] - \Delta t \sum_l \left(\sum_{p=1}^{n_p} h_p^l \rho_p^l \Gamma_p^l \Delta \psi^l + \Gamma_c^l \Delta T^l \right) + \Delta t \sum_{p=1}^{n_p} h_p \rho_p q_p = 0. \quad (5)$$

Here V is the control volume for which the system is being solved, $q_p = \bar{q}_p V$ 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 and Voskov, 2018b). 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 Eq. (4), and the pressure and enthalpy for the energy Eq. (5).

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:

$$J(\omega^n)(\omega^{n+1} - \omega^n) = -r(\omega^n), \quad (6)$$

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 Eqs. (4) and (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 details in Voskov (2017) and Khait and Voskov (2018a,b).

2.3. 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 Eq. (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:

$$Q_{ij} = \Gamma_{ij}^{ij} (P_i - P_j), \quad (7)$$

where:

Q_{ij}	flow rate at interface of cells i and j ,
Γ_{ij}^{ij}	phase transmissibility at interface of cells i and j ,
P_i	pressure of cell i ,
P_j	pressure of cell j .

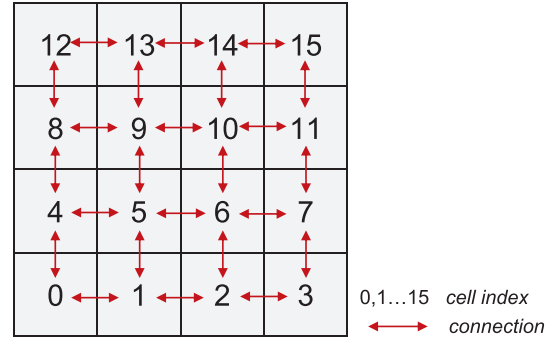


Fig. 1. 4x4 Cartesian grid denoting cell indexing and showing neighbouring connections. Indexing is based on a Cartesian structured mesh for simplicity.

Table 1

Connectivity list of the example grid from Fig. 1.

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

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:

$$Q_{ij}^h = \Gamma_c^{ij} (T_i - T_j), \quad (8)$$

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_{ij}^h is the heat flux at interface ij .

To evaluate the flux between two adjacent control volumes, a so-called *connectivity list* is constructed, i.e. for each interface between two neighbouring control volumes, the indices of these cells are listed together with the transmissibility (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.

Fig. 1 shows a simple example of a 2D Cartesian structured grid, with corresponding cell indexing. Table 1 shows its connectivity list. The list is expressed as two arrays, cell i and cell j , where each column represent a connection pair. Each pair has an associated interface transmissibility stored in the connectivity list.

3. Multi-level grid generation

The adaptivity of the grid in the developed AMR scheme is based on a hierarchical representation of connectivity list. The simulation grid is composed of several predefined *levels* representing the same geological properties at different resolutions. 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 4 and 5) for each mesh element, and a list of connectivity with corresponding transmissibility between neighbouring cells.

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

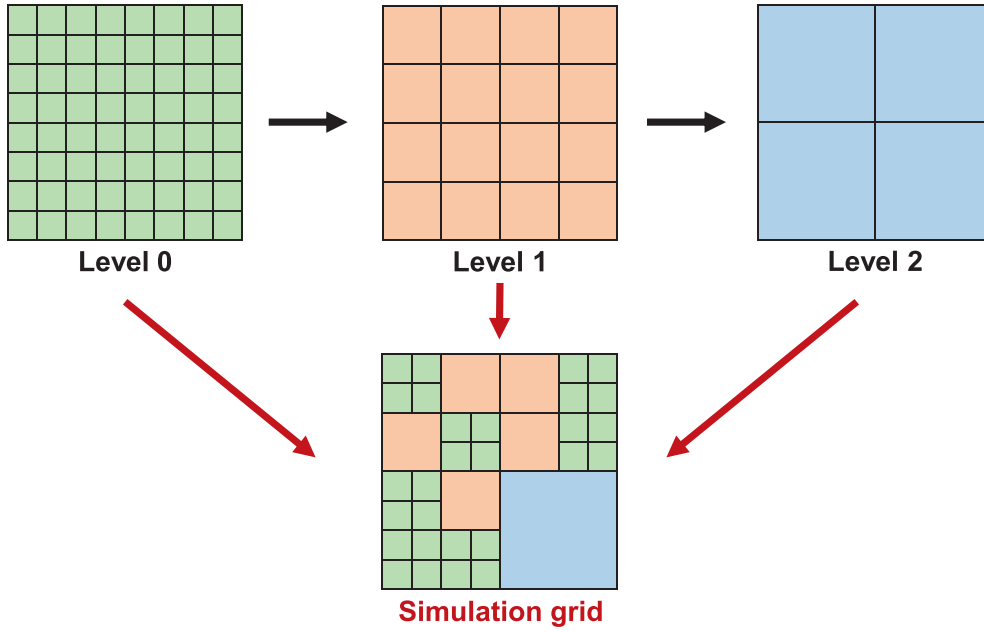


Fig. 2. 2D Multi-level grid with three pre-constructed grids (levels) with an example simulation grid which is constructed by aggregating control volumes from different levels.

of cells from grid-level $(n - 1)$, resulting in a hierarchical relationship (Karimi-Fard and 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 Fig. 2.

3.1. Cell aggregation

A mesh consists of a set of finite control volumes, each having vertices with allocated coordinates. To conduct cell aggregation, the centroid is first computed for each mesh element within the grid. Fig. 3 shows an example 2D unstructured grid 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 $(\frac{x_1+x_2+x_3}{3}, \frac{y_1+y_2+y_3}{3})$, where x_i and y_i are the coordinates of the vertices. Each mesh element 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 Δx and Δy using a predefined scaling factor. 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 :

$$\text{if } i \leq x_{cf} < i + \Delta x \text{ and } j \leq y_{cf} < j + \Delta y, \text{ cell } f \in \text{cell } F. \quad (9)$$

Fig. 3 shows the range partitioning (illustrated by the white lines) for a 2D unstructured grid. 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 coordinates fall within this plane, are aggregated to form one coarse cell. For example, cells 41, 46, 68, 77, 84, 92, 106, 111 and 118 form coarse cell 0.

For the given 2D unstructured grid example in Fig. 3, the so-called level 1 - i.e. the next level of coarsening - is shown in Fig. 4. 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.

For further steps into the generation 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 tabulates this list for the example above (Figs. 3 to 4). This type of list is generated for each coarse level (level > 0) in the hierarchical grid. These lists are stored for the construction of the cell properties (e.g. volume, porosity)

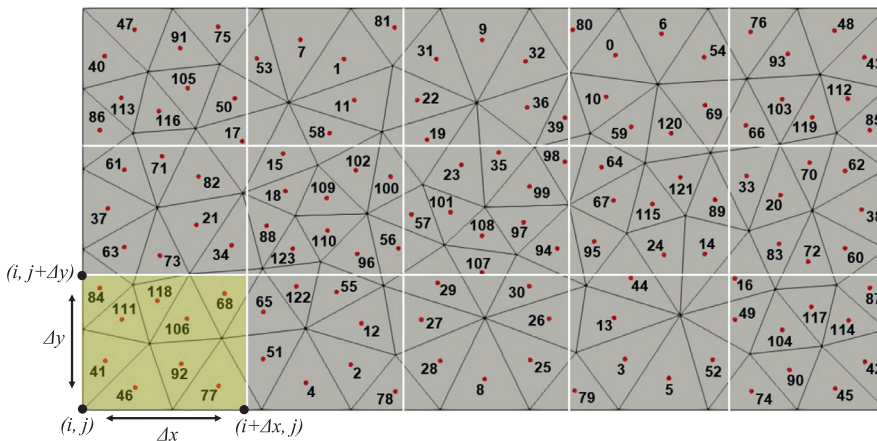


Fig. 3. 2D unstructured grid with centroids and with range partitioning (represented by the white lines) in the x - and y -direction with Δx and Δy spacing respectively. Aggregation is carried out for cells whose centroid fall within a given x - ($[i : i + \Delta x]$) and y -range ($[j : j + \Delta y]$). E.g., all cells whose centroids are found within the yellow-highlighted 2D range are aggregated to form one coarse cell.

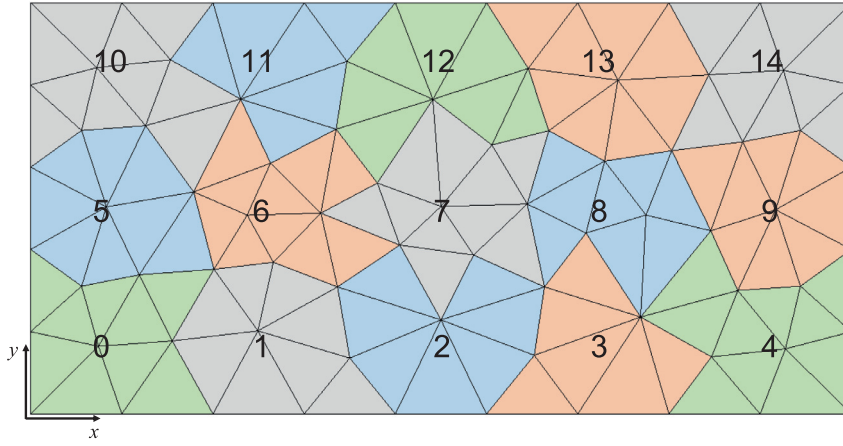


Fig. 4. 2D unstructured grid of Fig. 3 after cell aggregation. This grid represents the next level of coarsening of the finest grid: level 1.

Table 2

"fines_in_coarse" list of the example 2D unstructured grid of Figs. 3 - 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

of the coarse levels, where the cell data from the fine level is needed during upscaling.

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 and Durlofsky (2014).

After cell aggregation is conducted, the connectivity list is then constructed describing all connections within each level and the inter-level connections. To illustrate the methodology, we use the simple structured grid from Fig. 1, where cell aggregation was performed to form one coarse level.

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:

$$\{\mathbf{x} | \mathbf{x} \in i\} \cap \{\mathbf{x} | \mathbf{x} \in j\}. \quad (10)$$

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

Table 3

Interfaces l contained in each cell i for level 0. The interfaces are expressed as connection pairs, describing the faces for each cell i .

Fine cell i	Faces f
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)

Table 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)

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.

For the construction of the coarse level connectivity list, we first store for each cell i , connection pairs (interfaces) which consist of cell i , describing its faces. Table 3 illustrates this methodology for the fine grid in Fig. 1.

A similar list is constructed for coarser levels (level > 0), which is constructed by aggregating the faces f of the fine grid cells i (listed in Table 3 in this example) contained in each coarse cell I . Inner fine interfaces are unaccounted for, as they are not contained in the coarse interface. For the example above (Fig. 5), this results in the list shown in Table 4.

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

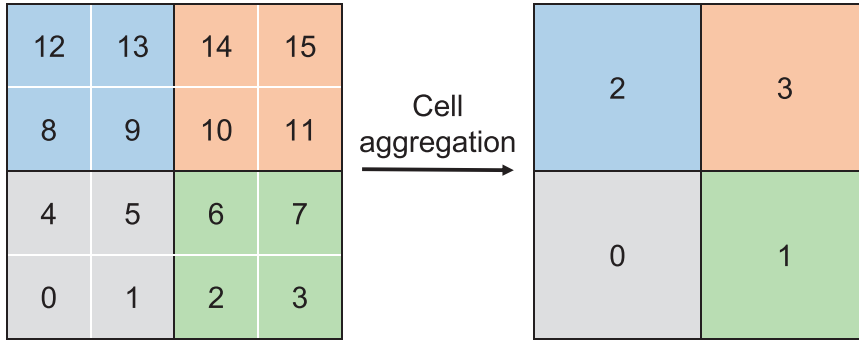


Fig. 5. 4x4 Cartesian grid showing cell aggregation, resulting in four coarse grid cells.

cell J . This is expressed as:

$$\{\mathbf{f}|\mathbf{f} \in I\} \cap \{\mathbf{f}|\mathbf{f} \in J\}. \quad (11)$$

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

$$\{\mathbf{f}|\mathbf{f} \in I\} \cap \{\mathbf{f}|\mathbf{f} \in i\}, \quad \text{where } \{i|i \notin I\} \quad (12)$$

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, which describe the full hierarchical grid.

3.2. Transmissibility and upscaling

In this work, the AMR method is implemented for unstructured grids of any geometry. The definition of the transmissibility for unstructured grids is expressed as:

$$\Gamma_p^{12} = \Gamma^{12} \lambda \quad \text{with} \quad \Gamma^{12} = \left(\frac{\alpha_1 \alpha_2}{\alpha_1 + \alpha_2} \right) \quad \text{and} \quad \alpha_i = A \frac{k_i}{D_i} n \cdot d_i, \quad (13)$$

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 $[k_x, k_y, k_z]$ coordinates multiplied by the unit vector d_i .

To solve the mass conservation Eq. (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^I must be computed to approximate thermal conductive flux in the energy Eq. (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 I divided by the sum of the distances D_1 and D_2 from centroids to interface I , multiplied by the average conduction κ_{12} :

$$\Gamma_c^{12} = \kappa_{12} \frac{A}{D_1 + D_2}. \quad (14)$$

As mentioned earlier, level 0 is represented by an array of volumes, depths 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 ;

$$V_I = \sum_{i \in I} v_i. \quad (15)$$

Depth upscaling is done by taking the average of the fine scale depths. 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 ;

$$\bar{\phi}_I = \frac{1}{V_I} \sum_{i \in I} v_i \phi_i. \quad (16)$$

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 and 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 by rearranging the flow Eq. (7). These approaches can be applied to unstructured coarse grids with generally-shaped control volumes (Karimi-Fard and Durlofsky, 2014). The coarsening technique defines the coarse transmissibility Γ_p^{IJ} between two adjacent control volumes I and J . This is expressed as:

$$\Gamma_p^{IJ} = \left| \frac{Q_{IJ}}{P_I - P_J} \right|. \quad (17)$$

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:

$$P_I = \frac{1}{V_I} \sum_{i \in I} v_i p_i, \quad P_J = \frac{1}{V_J} \sum_{j \in J} v_j p_j, \quad (18)$$

$$Q_{IJ} = \sum_{(i \in I, j \in J)} Q_{ij} = \sum_{(i \in I, j \in J)} \Gamma_p^{ij} (p_i - p_j),$$

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 . For inter-level connections, a similar approach is

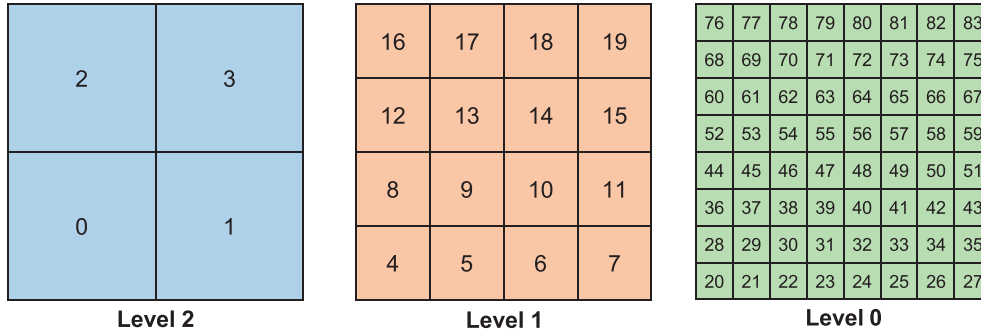


Fig. 6. Example of a Cartesian structured 3-level hierarchical grid with global indexing. Numbering starts at the coarsest level and finishes at the finest level of refinement.

used. For a given fine cell i and coarse cell J with interface iJ , Eq. (18) is used with $P_I = p_i$, the pressure of fine cell i , and P_J the pressure of coarse cell J . 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 Eq. (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.

3.2.1. Global indexing

Once all needed parameters at every hierarchical level are evaluated, which include cell properties and 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 Fig. 6.

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 Fig. 6, 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 array of cell properties, global indexing is applied by simply concatenating the arrays 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 array 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.

4. Dynamic adaptivity framework

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,

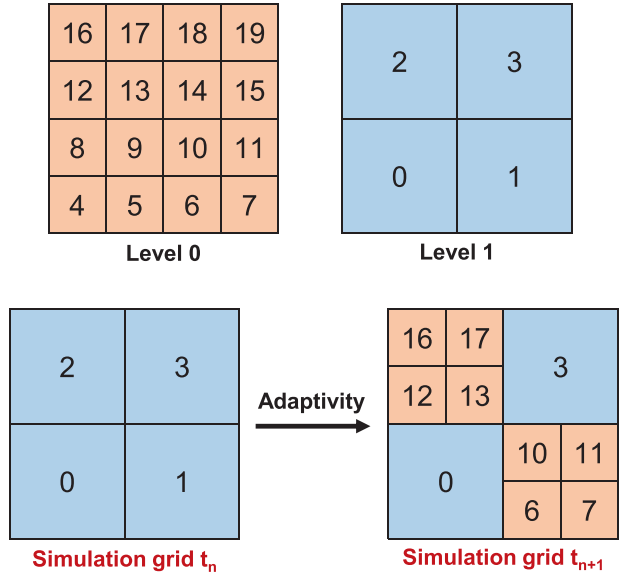


Fig. 7. Schematic representation of a 2-level hierarchical grid, with illustrated the adaptivity procedure and the redefinition of the active blocks for the simulation grid of the next time step t_{n+1} .

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 array 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. Fig. 7 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 array 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. As can be seen cell adaptivity results in an unstructured indexing..

Once the simulation grid is redefined and the array 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 array of volume, depth

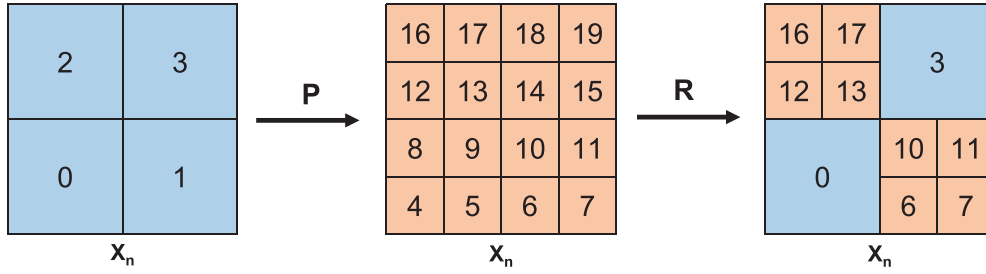


Fig. 8. Schematic representation of the prolongation and restriction for the example in Fig. 7.

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 Eqs. (4) and (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 :

$$X_i = X_I, \quad \forall i \in I. \quad (19)$$

Subsequently, restriction to the new simulation grid is conducted on the prolonged 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 and Durlof-sky, 2014):

$$X_I = \frac{1}{V_I} \sum_{i \in I} v_i X_i. \quad (20)$$

A schematic representation of this procedure for the 2-level hierarchical grid and for the new simulation grid of Fig. 7 (t_{n+1}) is shown in Fig. 8.

The model, however, necessitates sequential numbering for mesh generation. It can be seen in Fig. 7 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 global indexing is stored in a so-called *Global to Local* array for conversion back to the global indices for adaptivity check in the next time step. The described procedure, which redefines the grid configuration for the next simulation, is repeated at each time t . It is also important to note that all previously computed operators in the OBL method are re-used after each successive timestep. This is possible since the parameter space for each state dependent operator in the OBL method is decoupled from any spatial property or discretization. This provides a significant speeds-up of the computation especially when simulation property is expensive to evaluate.

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.

5. Applications for 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 (1) and energy (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 2 synthetic geothermal examples:

- A homogeneous reservoir with unstructured meshing,
- A heterogeneous fluvial system from Shetty et al. (2018) with low net-to-gross ratio.

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:

$$\begin{cases} \text{if } \Delta h_{ij} > 70 \text{ kJ,} & \text{mark cells i and j for refinement,} \\ \text{if } \Delta h_I < 40 \text{ kJ,} & \forall I \in I, \text{ mark cells } \{i \in I\} \text{ for coarsening.} \end{cases} \quad (21)$$

This adaptivity criteria is a simple heuristic and serves a practical purpose in this work. The proposed AMR method would greatly benefit from a more sophisticated criteria, for example, a criteria based on a posteriori error estimates similar to Vohralík and Wheeler (2013); Vohralík and Yousef (2018).

The geothermal examples are illustrated by showing the fine-scale solution at different time steps versus the AMR solution and the coarse-scale solution. Each synthetic example was analyzed quantitatively by conducting an error analysis 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. Moreover, to define the performance of the AMR method in terms of computational resources, the percentage of grid cells utilized in the simulation using the AMR grid, relative to the total number of cells in the fine-scale model was plotted for each example.

5.1. Case 1: homogeneous model

The first model is a simple 2D homogeneous reservoir (constant permeability) with unstructured triangular mesh. 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. Fig. 9 illustrates both levels, along with the permeability field (constant permeability of 2000 mD), and the well locations. The simulation parameters for this model are specified in Tables 6 and 7 in the Appendix.

The level 1 is illustrated in Fig. 9 where each color represents a coarse cell. As can be seen, cell aggregation was 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 Fig. 10. Figure (a) represents the temperature solution at fine scale, figure (b) the solution on the AMR grid, figure (c) shows the

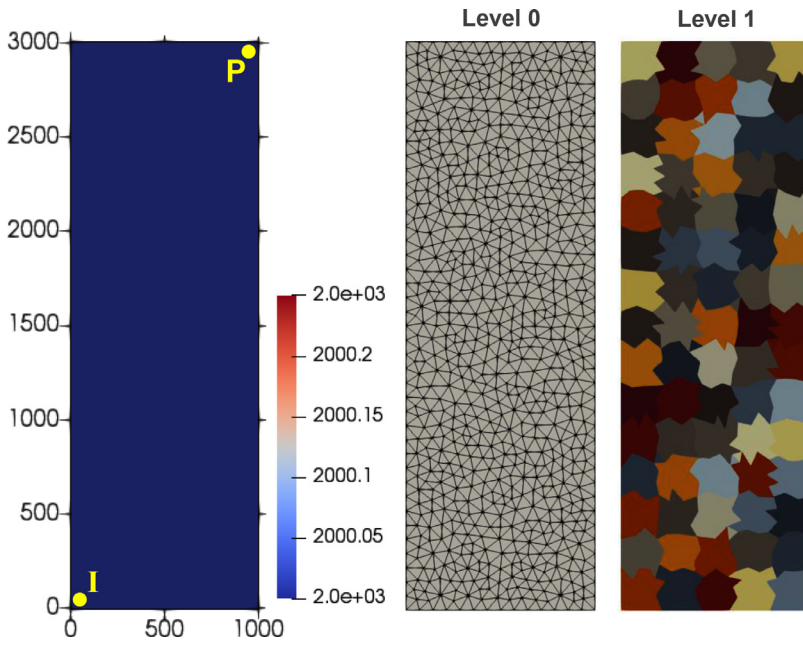


Fig. 9. Hierarchical grid of the unstructured homogeneous model. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement, with unstructured gridding; right figure: coarser level, level 1 where each color represents a coarse cell.

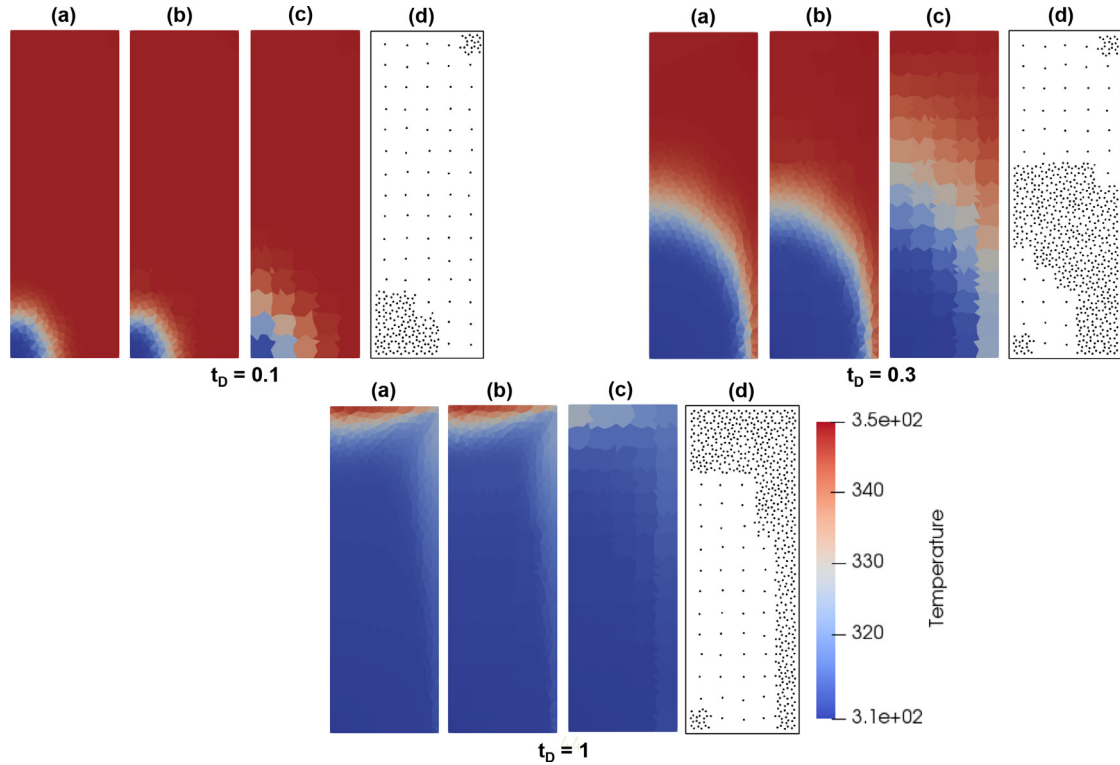


Fig. 10. Temperature solution of the homogeneous reservoir with unstructured gridding at $t_D = 0.01, 0.3$ and 1 . (a) represents the fine-scale solution; (b) represents the AMR solution; (c) is the coarse-scale solution; (d) is the node distribution of the AMR grid.

coarse-scale solution, and figure (d) shows the node distribution for the AMR simulation run.

The solution on the AMR grid demonstrates a particularly good match with the fine-scale solution. 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 .

The relative error of the AMR solution is significantly lower than the coarse solution in both the L2 and L-infinity norm (Fig. 11). Moreover, the number of cells is considerably reduced (see Fig. 12), 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. This considerably improves the performance of

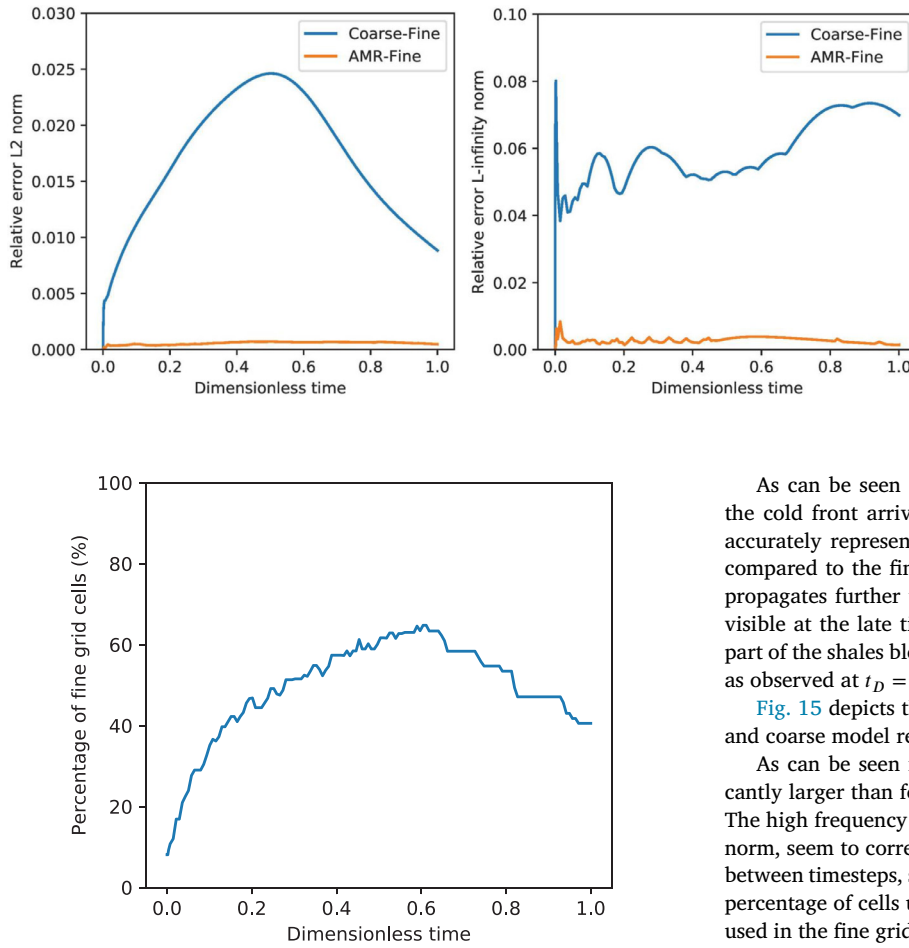


Fig. 12. Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the homogeneous model from Figs. 9 to 10.

simulation since the AMR approach is much more favourable in terms of efficient use of computational resources (see Table 5 in section 6).

5.2. Case 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. For geothermal applications, they represent an important source of heat for thermal recharge of the cold water front. Modelling of the shales however 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, with in total, a 5 by 6 shale block configuration. 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 Fig. 13. Level 0 consists of 4588 fine cells. Level 1 is constructed differently from the 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. The simulation parameters for this model are specified in Tables 6 and 7 in the Appendix.

Fig. 14 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.

Fig. 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 homogeneous model with unstructured gridding from Figs. 9 to 10.

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 $t_D = 0.7$. When the cold front passes part of the shale blocks and these have cooled down, coarsening occurs as observed at $t_D = 0.7$.

Fig. 15 depicts the error distribution through time of both the AMR and coarse model relative to the fine model.

As can be seen in Fig. 15, the error of the coarse model is significantly larger than for the AMR model, where the error is close to zero. The high frequency changes in the error, especially observed in the L_∞ norm, seem to correlate with refinement and coarsening of the mesh in between timesteps, similar to what was observed in Berrone (2010). The percentage of cells used in the AMR grid relative to the number of cells used in the fine grid is shown in Fig. 16. 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.

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 (see Table 5 in section 6).

5.3. Case 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 5 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 4×4 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 Fig. 17. The location of the injector (I) and producer (P) are depicted in yellow on the permeability distribution. The simulation parameters for this model are specified in Tables 6 and 7 in the Appendix.

The simulation was conducted until cold water breakthrough reached the producing well. Fig. 18 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.

The AMR mesh exhibits a significant improvement in temperature solution compared to the solution on the coarse grid. Refinement is mainly

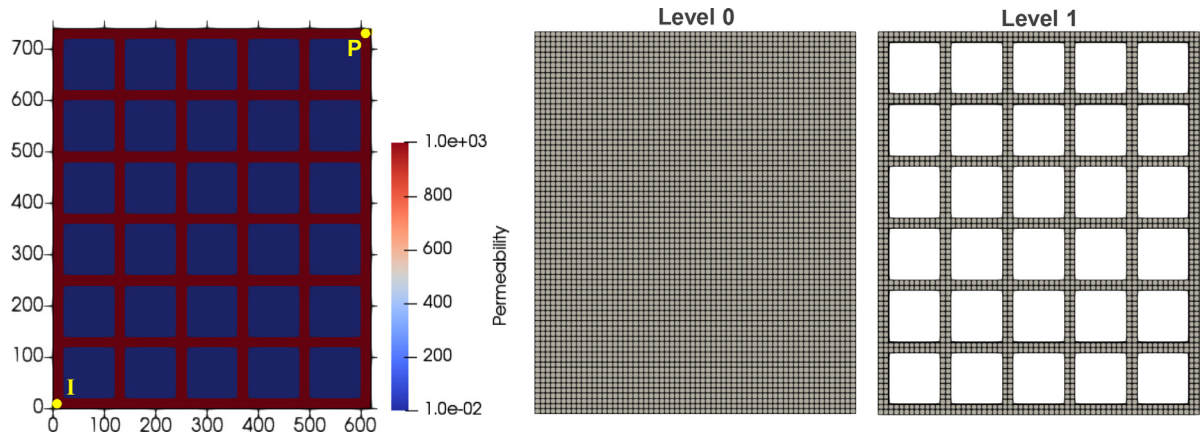


Fig. 13. Hierarchical grid of the model with shale blocks. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0; right figure: coarser level, level 1.

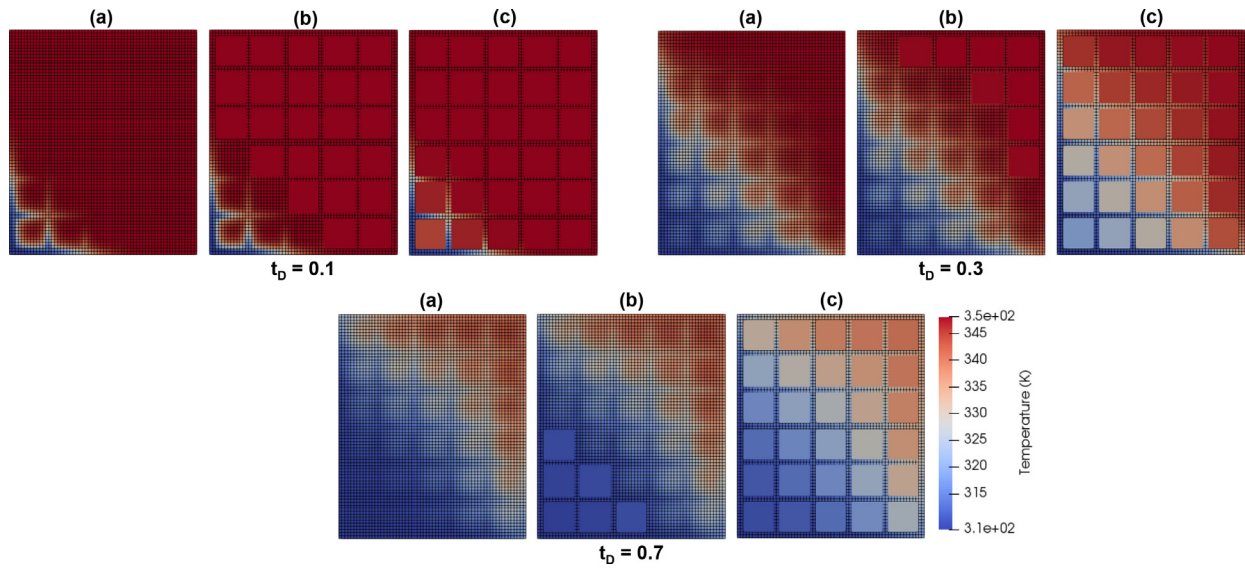


Fig. 14. Temperature solution of the shale blocks model at three different times: $t_D = 0.1$, 0.3 and 0.7 . (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

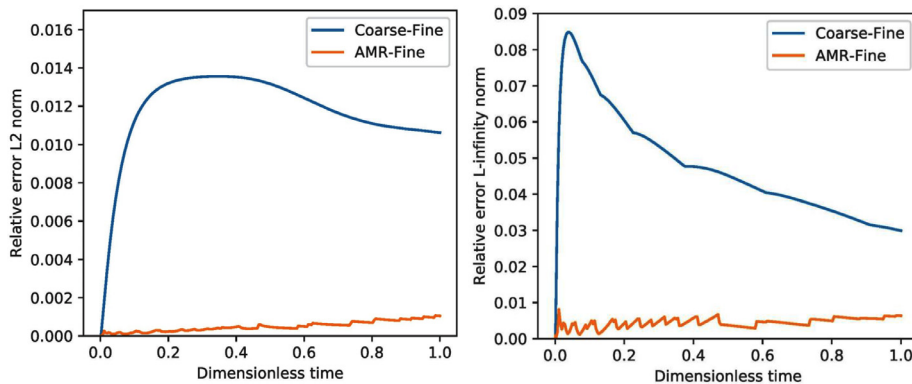


Fig. 15. L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the shale blocks model from Figs. 13 to 14.

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 be-

tween the coarse and AMR model. Fig. 19 shows the L2 norm and the L-infinity norm error in time.

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.

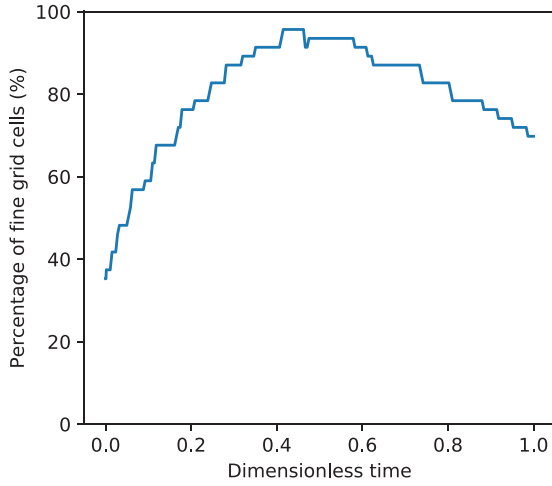


Fig. 16. Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the shale blocks model from Figs. 13 to 14.

The number of grid-cells used in the simulation ranges from 8 to 70% throughout the simulation (see Fig. 20). This represents a significant improvement in computational effort and time, while still capturing important features (see Table 5 in section 6).

5.4. Case 4: 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:



The kinetic reaction rate for this reaction is

$$r_k = AK_k \left(1 - \frac{Q}{K_{sp}}\right) s_s, \quad (23)$$

where A is the mineral surface area, K_k is the kinetic reaction constant, Q is the ion activity product, K_{sp} is the equilibrium product, and s_s is the solid saturation. Permeability is updated using a power-law relationship defined as follows

$$k = k_0 \left(\frac{\phi}{\phi_0}\right)^n, \quad (24)$$

where k_0 and ϕ_0 are the initial permeability and porosity, and n is the power-law exponent.

The proposed model 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 Fig. 21 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 Fig. 21. The simulation parameters for this model are specified in Tables 8 and 9 in the Appendix.

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, $x[\text{CaCO}_3]$. The adaptivity criteria used in this example are defined

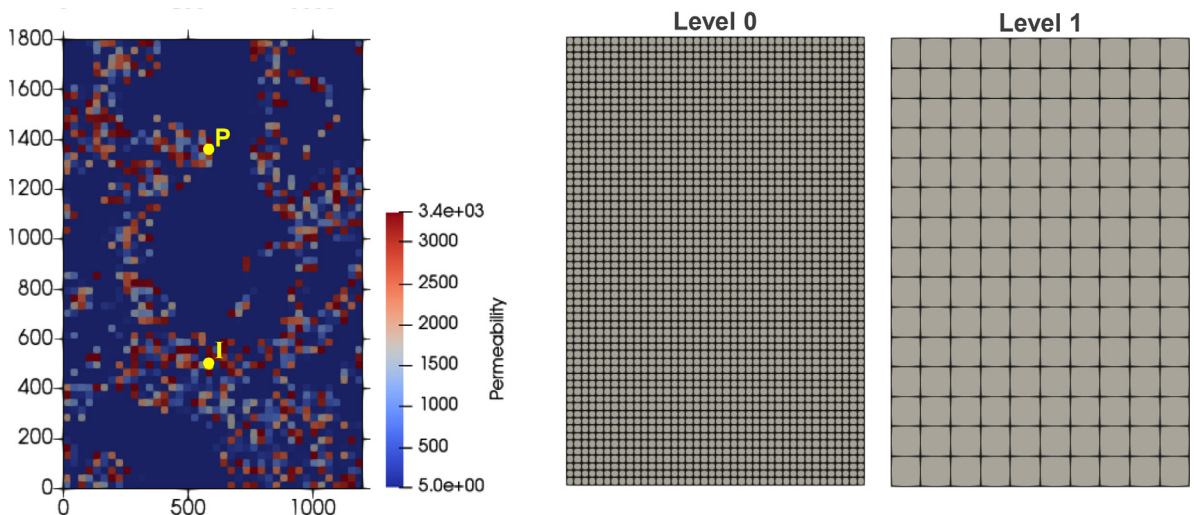


Fig. 17. Hierarchical grid of the heterogeneous model with low net-to-gross ratio. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

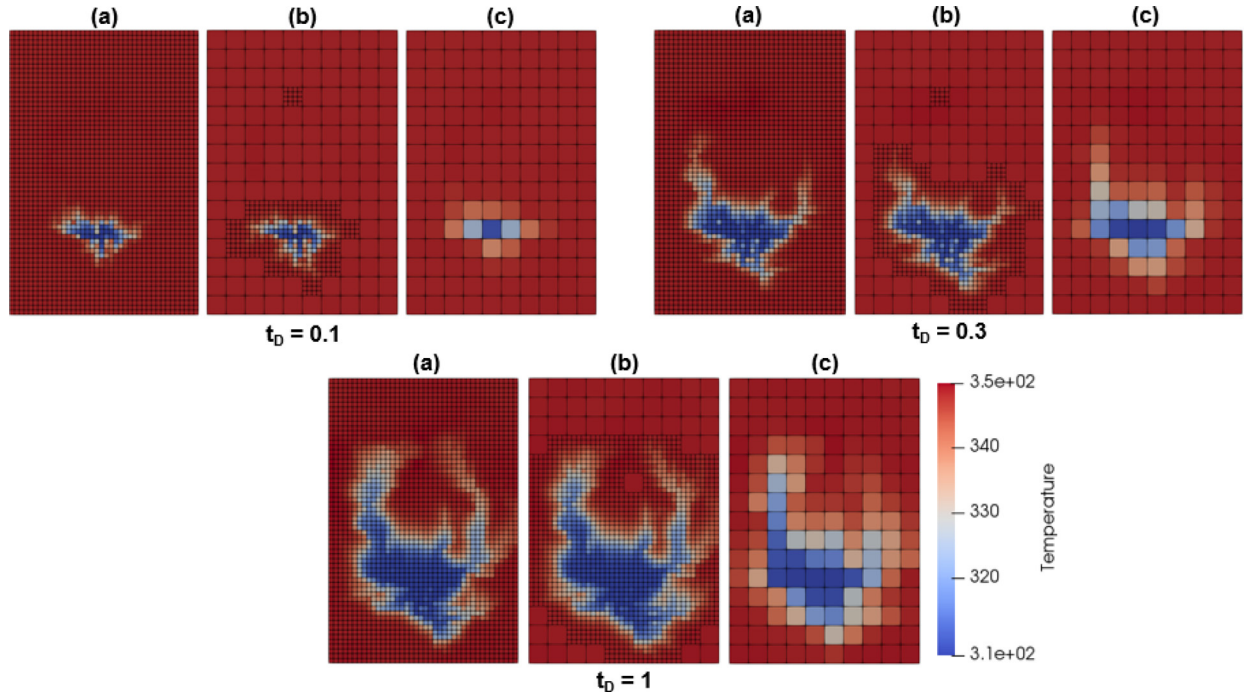


Fig. 18. Temperature solution of the heterogeneous model with low net-to-gross ratio at three different times: at $t_D = 0.1, 0.3$ and 1 . (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

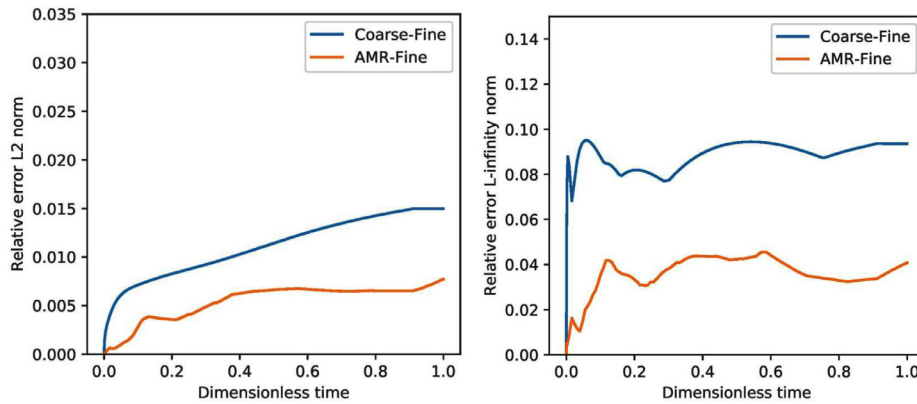


Fig. 19. L2 norm (left) and L-infinity norm (right) of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the heterogeneous model with low net-to-gross ratio from Figs. 17 to 18.

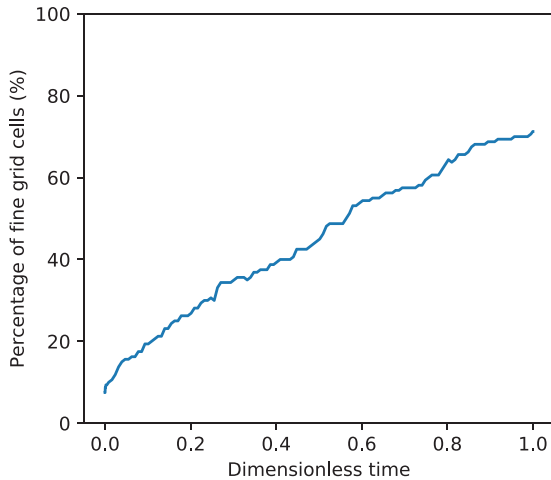


Fig. 20. Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the heterogeneous model with low net-to-gross ratio from Figs. 17 to 18.

as follows:

$$\begin{cases} \text{if } \Delta x[CaCO_3]_{ij} > 0.02, & \text{mark cells } i \text{ and } j \text{ for refinement,} \\ \text{if } \Delta x[CaCO_3]_l < 0.01, & \forall l \in I, \text{ mark cells } \{v_i \in I\} \text{ for coarsening,} \end{cases} \quad (25)$$

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). Fig. 22 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.

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 mi-

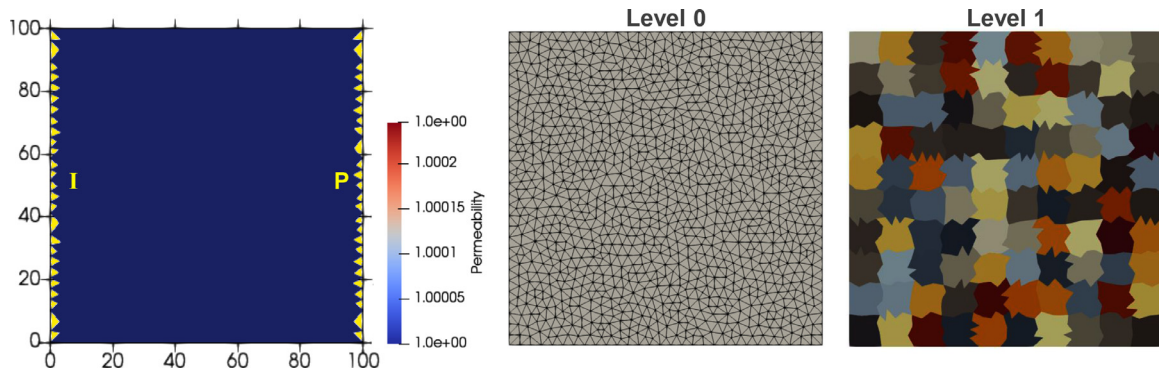


Fig. 21. Hierarchical grid of the wormhole model. Left figure: permeability field with reservoir dimensions and well locations; middle figure: level 0, the finest level of refinement; right figure: coarser level, level 1.

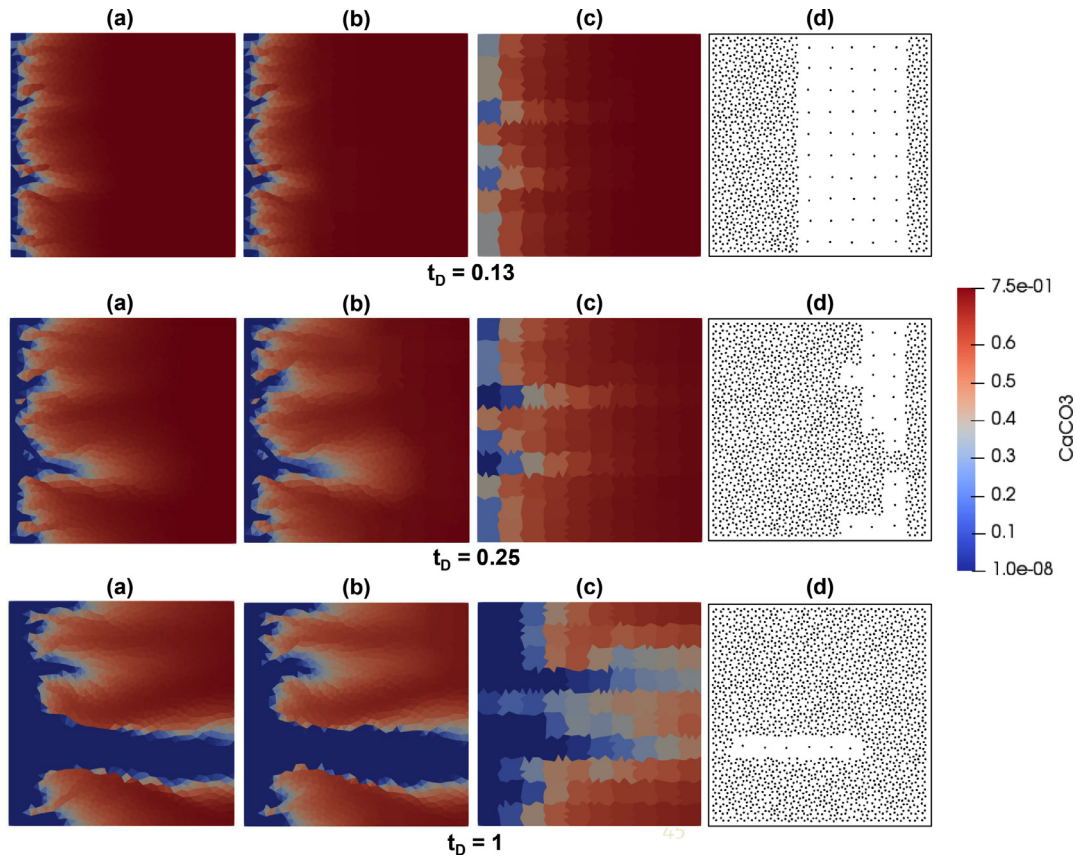


Fig. 22. Solid composition solution of the wormhole model at three different times. (a) is the fine-scale solution (level 0); (b) is the AMR solution; (c) is the coarse-scale solution (level 1).

nor 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, AMR and coarse-scale solutions, 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 Fig. 23 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.

Similarly to the previous example, we have analyzed the total number of cells used in the AMR model, relative to the total number of cells contained in the fine-scale grid. The graph in Fig. 24 shows this quantity expressed in percentage.

As can be seen, the number of cells used during the AMR simulation is overall less than the number of cells present in the fine-scale model. Initially, the number of cells starts at 20%, which represents the use of

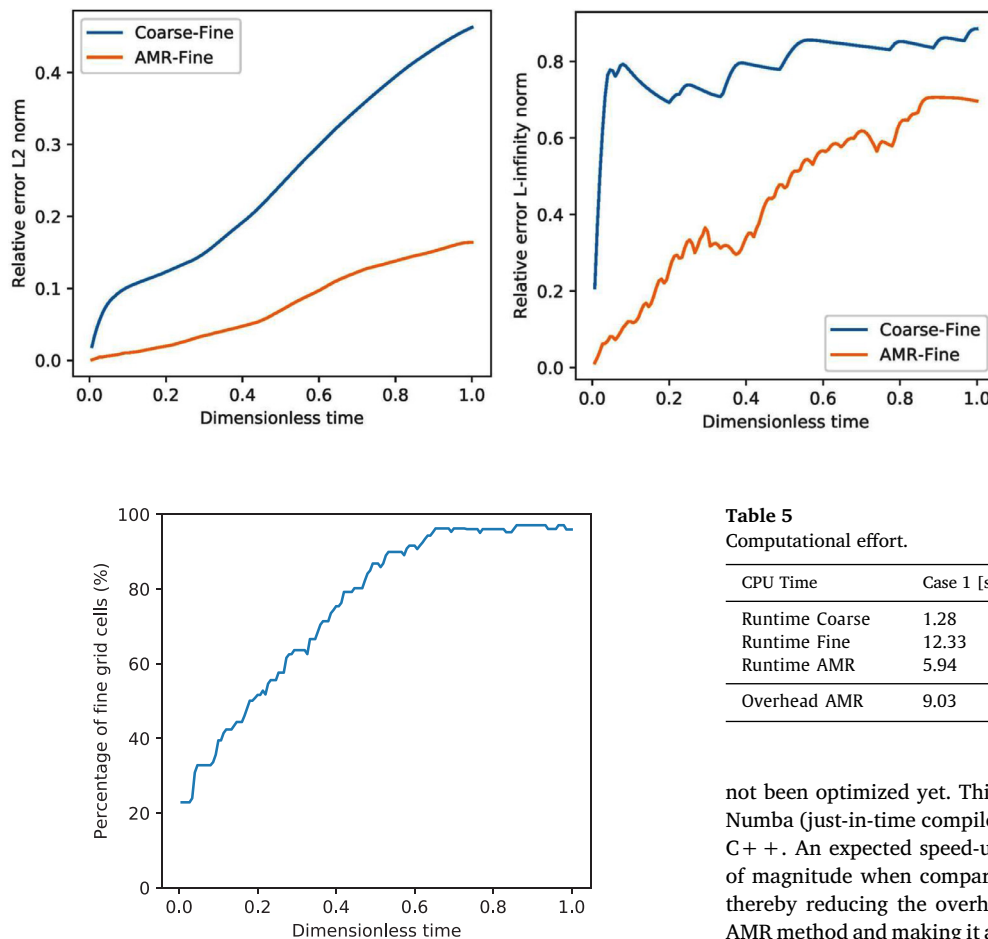


Fig. 23. L2 norm and L-infinity norm of the difference between the coarse model and the fine model, and between the AMR model and the fine model, both relative to the fine-scale solution, for the wormhole model from Figs. 21 to 22.

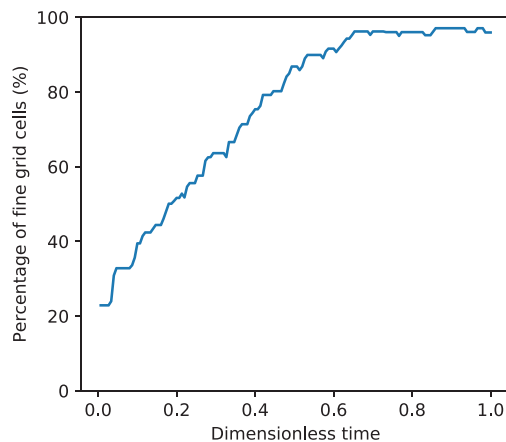


Fig. 24. Percentage of mesh elements used during the simulation of the AMR model, relative to the total number of cells in the fine-scale model in time for the wormhole model from Figs. 21 to 22.

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 number of cells starts to decrease, which indicates coarsening at some locations. See Table 5 in the next section for the actual computational time. 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.

6. Discussion

In the results section, all computational speed-up was indicated in terms of % of active cells w.r.t. fine-scale. In this section, the actual CPU times are highlighted for all cases. In order to have a fair comparison, the same nonlinear solver (Newton's based update with a fixed number of iterations), linear solver (direct one) and time-stepping strategy is used for the fine, coarse, and AMR runs. All these results are shown in Table 5.

Even though the overhead of the AMR method is substantial, this can easily be explained by the non-vectorized Python implementation of the AMR procedure vs highly optimized Python and C++ implementation of the conventional simulation used in the coarse and fine simulation. Since the scope of this work is a proof of concept of the proposed AMR procedure prototyped outside of the simulation loop, our AMR code has

Table 5
Computational effort.

CPU Time	Case 1 [s]	Case 2 [s]	Case 3 [s]	Case 4 [s]
Runtime Coarse	1.28	1.70	1.60	13.61
Runtime Fine	12.33	4.35	21.73	167.33
Runtime AMR	5.94	3.06	8.87	91.85
Overhead AMR	9.03	36.33	15.54	321.86

not been optimized yet. This can be solved by either an application of Numba (just-in-time compiler for Python) or rewriting the procedure in C++. An expected speed-up, in our experience, is around two orders of magnitude when compared to the original Python implementation, thereby reducing the overhead to around 1.5% of the runtime of the AMR method and making it a viable strategy for geoscience applications.

The framework presented in this paper is developed in the DARTS platform which can be used for a more general set of applications related to the energy transition. However, it is important to note the major differences in various energy applications. For example, in two types of applications shown in this study (geothermal and chemical dissolution cases), the coarser representation is still capable of accurately capturing important features of the geothermal dynamic behaviour. The coarse-scale simulation in chemical dissolution, however, completely fails to represent the same dynamic behaviour (dissolution pattern) and effective characterization of the process (e.g. effective rock dissolution). It is therefore evident, as is also pointed out in the literature, that problems contained localized sharp gradients can greatly benefit from the AMR technique.

7. Conclusions

This study aimed at developing an Adaptive Mesh Refinement (AMR) technique in *Delft Advanced Research Terra Simulator* (DARTS) 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 an array 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. However, this strategy can easily be changed and improved.

The aggregation of the subdomains to form a coarser level is stored as an array of indices for the 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 up-scaling is applied in order to acquire the transmissibility of coarser levels and the inter-level transmissibility. Each control volume has defined parameters that are relevant for solving the system (volume, porosity, depth etc).

Once the hierarchy of levels is complete, the simulation can be started. 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 prolonged 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 geothermal applications. Two models were tested, including a homogeneous model with unstructured gridding, a synthetic sugar-cube-like model with high permeability channels surrounded by shale blocks and a heterogeneous fluvial system model with a low net-to-gross ratio. High levels of solution accuracy relative to the reference fine-scale results are observed for both 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.

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 other applications implemented in DARTS. 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.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Stephan de Hoop: Methodology, Validation, Writing - review & editing. **Elodie Jones:** Software, Investigation, Writing - original draft. **Denis Voskov:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Acknowledgements

We would like to thank the entire DARTS group for their assistance and help. We would also like to acknowledge financial support from the SFB1313 research group at [University of Stuttgart](#) and particularly like to thank Beatrix Becker and Prof. Rainer Helmig for their valuable time and help.

Appendix

Table 6

Boundary conditions geothermal models (section 5.1, 5.2, and 5.3).

Parameter	Value
Rock heat conduction, κ_r [kJ/m/day/K]	200
Rock heat capacity, C_r [kJ/m ³ /K]	2500
Initial pressure, p_0 [bar]	100
Initial temperature, T_0 [K]	348.15
Injection pressure, p_{inj} [bar]	150
Injection temperature, T_{inj} [K]	308.15
Production pressure, p_{prod} [bar]	60

Table 7

Reservoir and simulation parameters geothermal models (section 5.1, 5.2, and 5.3).

Parameter	Value (5.1)	Value (5.2)	Value (5.3)
Sand permeability, k_{sa} [mD]	2000	1000	3000 ± 12.5%
Sand porosity, ϕ_{sa} [-]	0.25	0.25	0.3 ± 12.5%
Shale permeability, k_{sh} [mD]	N/A	1e-2	5
Shale porosity, ϕ_{sh} [-]	N/A	0.001	0.1
Length domain, L_x [m]	1000	600	1200
Width domain, L_y [m]	3000	700	1800
Simulation time, t [days]	5500	1200	7600
Control volumes level 0 [-]	1420	4588	2400
Control volumes level 1 [-]	75	1618	150

Table 8

Boundary conditions chemical model (section 5.4).

Parameter	Value
Mineral surface area, A [m ²]	1
Kinetic reaction constant, K_k [kmole/day]	1e-3
Equilibrium product, K_{sp} [-]	0.24
Power-law exponent, n [-]	3
Initial pressure, p_0 [bar]	95
Initial composition, $z_{c,0}$ [Ca ²⁺ + CO ₃ ²⁻ , CaCO ₃ , H ₂ O]	[0.24, 0.75, 0.01]
Injection rate, q_{inj} [m ³ /day]	0.05
Injection composition, $z_{c,inj}$ [Ca ²⁺ + CO ₃ ²⁻ , CaCO ₃ , H ₂ O]	[0.01, 0.0, 0.99]
Production pressure, p_{prod} [bar]	65

Table 9

Reservoir and simulation parameters chemical model (section 5.4).

Parameter	Value
Initial permeability, k_0 [mD]	1 ± 5%
Initial porosity, ϕ_0 [-]	0.26
Length domain, L_x [m]	100
Width domain, L_y [m]	100
Simulation time, t [days]	2e6
Control volumes level 0 [-]	2194
Control volumes level 1 [-]	100

References

- Babaei, M., Nick, H., 2019. Performance of low-enthalpy geothermal systems: Interplay of spatially correlated heterogeneity and well-doublet spacings. *Appl. Energy* 253. <https://doi.org/10.1016/j.apenergy.2019.113569>.
- Bahrainian, S.S., Dezfouli, A.D., 2014. A geometry-based adaptive unstructured grid generation algorithm for complex geological media. *Comput. Geosci.* 68, 31–37.
- Berrone, S., 2010. Skipping transition conditions in a posteriori error estimates for finite element discretizations of parabolic equations. *ESAIM: Math. Modell. Numer. Anal.-Modélisation Mathématique et Analyse Numérique* 44 (3), 455–484.
- Berrone, S., Borio, A., Vicini, F., 2019. Reliable a posteriori mesh adaptivity in discrete fracture network flow simulations. *Comput. Methods Appl. Mech. Eng.* 354, 904–931.
- Branets, L.V., Ghai, S.S., Lyons, S.L., Wu, X.-H., 2009. Challenges and technologies in reservoir modeling. *Commun. Comput. Phys.* 6 (1), 1.
- Chen, M., Tompson, A., Mellors, R., Abdalla, O., 2015. An efficient optimization of well placement and control for a geothermal prospect under geological uncertainty. *Appl. Energy* 137, 352–363. <https://doi.org/10.1016/j.apenergy.2014.10.036>.
- Christensen, J., Darche, G., Dechelette, B., Ma, H., Sammon, P., et al., 2004. Applications of dynamic gridding to thermal simulations. *SPE International Thermal Operations and Heavy Oil Symposium and Western Regional Meeting. Society of Petroleum Engineers*.
- Christie, M.A., 1996. Upscaling for reservoir simulation. *J. Petrol. Technol.* 48 (11), 1–004.
- Croijmans, R., Willems, C., Nick, H., Bruhn, D., 2016. The influence of facies heterogene-

- ity on the doublet performance in low-enthalpy geothermal sedimentary reservoirs. *Geothermics* 64, 209–219. <https://doi.org/10.1016/j.geothermics.2016.06.004>.
- Cui, G., Zhang, L., Ren, B., Enechukwu, C., Liu, Y., Ren, S., 2016. Geothermal exploitation from depleted high temperature gas reservoirs via recycling supercritical CO₂: Heat mining rate and salt precipitation effects. *Appl. Energy* 183, 837–852. <https://doi.org/10.1016/j.apenergy.2016.09.029>.
- Cusini, M., van Kruijsdijk, C., Hajibeygi, H., 2016. Algebraic dynamic multilevel (ADM) method for fully implicit simulations of multiphase flow in porous media. *J. Comput. Phys.* 314, 60–79.
- 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, 2024. Citeseer, 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 J.* 13 (01), 58–67.
- Jenny, P., Lee, S., Tchelepi, H.A., 2003. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *J. Comput. Phys.* 187 (1), 47–67.
- Kala, K., Voskov, D., 2020. Element balance formulation in reactive compositional flow and transport with parameterization technique. *Comput. Geosci.* 24 (2), 609–624. <https://doi.org/10.1007/s10596-019-9828-y>.
- Karimi-Fard, M., Durlofsky, L.J., 2012. Accurate resolution of near-well effects in upscaled models using flow-based unstructured local grid refinement. *SPE J.* 17 (04), 1–084.
- Karimi-Fard, M., Durlofsky, L.J., 2014. Unstructured adaptive mesh refinement for flow in heterogeneous porous media. *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 Resour. Res.* 42 (10).
- Khait, M., Voskov, D., 2018. Adaptive parameterization for solving of thermal/compositional nonlinear flow and transport with buoyancy. *SPE Journal* 23, 522–534. <https://doi.org/10.2118/182685-PA>.
- Khait, M., Voskov, D., 2018. Operator-based linearization for efficient modeling of geothermal processes. *Geothermics* 74, 7–18. <https://doi.org/10.1016/j.geothermics.2018.01.012>.
- Lim, K.-T., Schiozer, D., Aziz, K., 1995. A new approach for residual and jacobian arrays construction in reservoir simulators. *SPE Comput. Appl.* 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. *Comput. Geosci.* 16 (3), 577–592.
- Sammon, P.H., et al., 2003. Dynamic grid refinement and amalgamation for compositional simulation. *SPE reservoir simulation symposium*. Society of Petroleum Engineers.
- Shaik, A., Tomin, P., Voskov, D., 2018. Modeling of near-well matrix acidization. 43rd Workshop on Geothermal Reservoir Engineering.
- Shetty, S., Voskov, D., Bruhn, D.F., 2018. Numerical strategy for uncertainty quantification in low enthalpy geothermal projects. Workshop on Geothermal Reservoir Engineering.
- Trangenstein, J.A., 2002. Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Adv. Water Resour.* 25 (8–12), 1175–1213.
- 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. *IOR 2011-16th European Symposium on Improved Oil Recovery*.
- Vohralik, M., Wheeler, M.F., 2013. A posteriori error estimates, stopping criteria, and adaptivity for two-phase flows. *Comput. Geosci.* 17 (5), 789–812.
- Vohralik, M., Yousef, S., 2018. A simple a posteriori estimate on general polytopal meshes with applications to complex porous media flows. *Comput. Methods Appl. Mech. Eng.* 331, 728–760.
- Voskov, D.V., 2017. Operator-based linearization approach for modeling of multiphase multi-component flow in porous media. *J. Comput. Phys.* 337, 275–288.
- Wang, Y., Hajibeygi, H., Tchelepi, H.A., 2014. Algebraic multiscale solver for flow in heterogeneous porous media. *J. Comput. Phys.* 259, 284–303.
- Wang, Y., Voskov, D., Khait, M., Bruhn, D., 2020. An efficient numerical simulator for geothermal simulation: a benchmark study. *Appl. Energy* 264. <https://doi.org/10.1016/j.apenergy.2020.114693>.
- Willems, C., 2017. Doublet deployment strategies for geothermal hot sedimentary aquifer exploitation: application to the Lower Cretaceous Nieuwerkerk Formation in the West Netherlands Basin. Delft University of Technology.
- Willems, C., M. Nick, H., 2019. Towards optimisation of geothermal heat recovery: An example from the West Netherlands Basin. *Appl. Energy* 247, 582–593. <https://doi.org/10.1016/j.apenergy.2019.04.083>.