Society of Petroleum Engineers

SPE 182681-MS

A General Non-Uniform Coarsening and Upscaling Framework for Reduced-Order Modeling

K.-A. Lie, SINTEF; K. Kedia, ExxonMobil Upstream Research Company; B. Skaflestad, SINTEF; X. Wang, Y. Yang, X.-H. Wu, N. Hoda, ExxonMobil Upstream Research Co.

This paper was prepared for presentation at the SPE Reservoir Simulation Conference held in Montgomery, Texas, USA, 20-22 February 2017.

Abstract

This paper presents a general framework for constructing effective reduced-order models from an existing high-fidelity reservoir model, irrespective of grid topology. We employ a flexible hierarchical grid coarsening strategy that is designed to preserve geologic features and structures in the underlying model such as environments of deposition and faults. The strategy supports selecting and combining coarsening methods that are targeted to the flow patterns in different parts of the reservoir. This includes, but is not limited to, explicit user-imposed boundaries, using efficient field-wide flow indicators, topological and geometric partitioning and methods for amalgamating and splitting clusters of cells.

Collectively, these schemes enable an automatic strategy that separates a model into flow-dependent compartments that are respectively close to, far away from, or in between regions of sharp flow transients such as wells. These compartments may then be coarsened using different tailored techniques and target grid resolutions providing much more flexibility compared to traditional coarsening methods. We demonstrate that various techniques for flow-based transmissibility upscaling can be deployed on the resulting coarsened model to compute effective model properties. The hierarchical construction strategy allows efficient exploration of the geologic features of a reservoir that most impact flow patterns and well communication. The coarsened models are shown to be rank and trend accurate, enabling a more exhaustive sensitivity analysis if needed. We study the accuracy of the reduced-order model with a particular emphasis on the upscaled model's ability to capture effects of multiple phases in simulation runs compared to the full high-fidelity model.

Introduction

The modern process of characterizing and managing assets leads to creating reservoir models that are becoming increasingly large, detailed and complex. Accounting for heterogeneous structures and geologic processes on a variety of spatial scales, these models are used for a multitude of purposes including uncertainty quantification, history matching, and production optimization. Still, the computational challenge of full-field multiphase simulation on detailed models with fully unstructured grids is often unmet in the context of typical reservoir engineering workflows–even on contemporary high performance computers with state-of-the-art numeric methods and simulation software. To optimize recovery of oil and gas fields there is a growing need for more accurate and robust computational methods that can seamlessly move between different spatial and temporal resolutions and be used to reduce model fidelity (e.g., the number of degrees of freedom) to a level that is sufficient to resolve flow physics and satisfy requirements on computational costs, while preserving the important characteristics of the underlying static and dynamic data. Model rescaling should be as automated as possible to reduce the turnaround time of modeling. Today, however, workflows for model rescaling are usually manual and work-intensive.

To develop a reduced-order model from a high-resolution model, one must first create a grid model with reduced spatial resolution (grid coarsening) and then map reservoir properties to the new coarse grid using an appropriate upscaling method, see the reviews in, e.g., (Durlofsky 2003, 2005; Christie 1996; Farmer 2002; Stern 2005). To distinguish entities in the reduced and the original grid, we will use the word 'block' when referring to the reduced model and 'cell' when referring to the original high-resolution model. For corner-point and other types of grid that have an underlying Cartesian topology, the standard approach to coarsening is to merge layers in the vertical direction and/or reduce the number of cells in the lateral directions by a given factor. For grids that are unstructured laterally or in 3D, one must specify geometric constraints that determine the shape of the new grid blocks and the degree to which they align with the cells in the original grid. However, the literature also contains many examples of more advanced coarsening algorithms that define grid with spatially varying resolution by equilibrating a density

measure or by defining coarse blocks so that they minimize the variance of some indicator function within blocks and maximize it between blocks. Density measures and indicators can be defined using *a priori* quantities like petrophysical parameters; flow-based quantities like fluid velocity, vorticity, or streamlines; or statistical or *a posteriori* goal-oriented error indicators that measure how a particular point influences the error in production responses or other predefined quantities. Garcia et al. (1992) use permeability as the background indicator used to coarsen the grid, King (2007) reviews advances in grid coarsening based on static *a priori* properties, whereas King et al. (2006) explore *a priori* local error analysis to generate upscaled models by vertical combination of layers. Prevost et al. (2005) describe how to incorporate both grid-resolution targets and geometric constraints Evazi and Mahani (2010b) investigate Delaunay tessellation based on different grid-point density indicators, such as permeability variations, fluid velocity, and vorticity, whereas Branets et al. (2009) discuss constrained Delaunay and Voronoi griding with respect to the underlying reservoir geometry. In particular, flow-based coarsening has proved to be a powerful approach in combination with upscaling, and has been developed both for structured and unstructured grids. The basic goal of flow-based gridding is to introduce higher resolution in regions of high flow while allowing coarser resolution in regions of lower flow. Durlofsky et al. (1997) present a nonuniform approach that is constrained to be aligned with the underlying fine grid. Techniques based on streamlines are investigated by Verma and Aziz (1997); Cirpka et al. (1999); Castellini et al. (2000); Wen et al. (2003); He and Durlofsky (2006). Later research considers the use of vorticity to determine spatial resolution of coarse grids: Mahani and Muggeridge (2005) construct coarse grids that seek to preserve high vorticity regions; Ashjari et al. (2008) present a more rigorous approach for optimal coarse grids; and Evazi and Mahani (2010a) demonstrate a technique for generating unstructured coarse Voronoi grids.

Herein, we discuss a somewhat different approach that enables multilevel model reduction. This approach differs from the traditional grid coarsening/upscaling approach in the sense that the original high-resolution model is present at all times and the coarse grids is realized in the form of a partition vector defined over the original high-resolution grid. In other words, the coarse grid is represented as a partition vector in which each cell is assigned to a unique coarse block. These blocks will not have simple geometric forms but rather be complex polyhedral forms that follow the geometry of the original fine grid. This way, we are guaranteed to preserve reservoir volumes and avoid many subtle inconsistencies between cell and block geometries. From the point-of-view of a multiphase simulator, the partitioned grid will not be different from a standard grid, as both can be represented in terms of a connection graph having well-defined pore volumes associated with each vertex and transmissibilities associated with each edge.

Grid coarsening should in principle be formulated as a minimization problem, e.g., as done in standard graph-partitioning algorithms. Herein, however, we have focused on developing a set of ad hoc methods that give reasonable coarse grids that adapt to features in the geologic model. In particular, we have developed methods for generating non-uniform coarse grids that are constrained to a hierarchy of geologic features. It is well known that structural and stratigraphic frameworks have the largest impact on flow patterns, and preserving key concepts of these frameworks all the way to flow simulation is crucial to be able to reliably predict flow patterns in the reservoir (Branets et al. 2015). Each geologic feature (e.g., faults, subzones, environments of deposition, layering, depositional and lithographic facies, etc) in a reservoir model may have a unique impact upon the fluid flow. The difference in the relative impact of the geologic features on performance predictions enables us to identify critical features first, incorporate additional data as it becomes available, and better judge when to stop trying to resolve the uncertainty associated with a particular feature.

The new framework is very general and contains a wide variety of heuristic algorithms that will automatically or semi-automatically produce fit-for-purpose grids of sufficient quality. In particular, the framework will encompass a large number of different coarsening techniques that we have developed over the years for three different purposes: (i) to increase the accuracy of multiscale pressure solvers by defining grids that adapt to underlying features in a geo-cellular model Aarnes et al. (2006, 2008); Natvig et al. (2011, 2012); Alpak et al. (2012); (ii) to increase the accuracy of upscaling methods by defining grids that avoid upscaling features in the geologic model that are distinctly different; and (iii) to find suitable (flow-adapted) coarse grids that best resolve fluid transport and minimize loss of accuracy when representing saturations on a coarser scale Aarnes et al. (2007); Hauge et al. (2010, 2012); Hauge (2010).

The main purpose of the paper is to outline the general framework, present a few specific coarsening algorithms that have been implemented in an in-house simulator, and study the accuracy of the resulting reduced-order models. The interested reader may also explore the `agglom` module of the open-source Matlab Reservoir Simulation Toolbox MRST (2016); Lie (2015), which implements a variety of algorithmic components that easily can be combined to generate various types of coarse partitions that adapt to geologic features and flow patterns.

Multilevel Hierarchical Coarsening

We consider coarsening of a fine grid defined over a singly-connected domain Ω. The grid is assumed to consist of a set of grid cells, $\{c_i\}_{i=1}^n$. No assumptions are made on the geometry of the grid, whereas for the topology we assume that there is an explicit mapping $\mathcal{N}(c_i)$ between cell c_i and its nearest neighbors c_ℓ. On top of this fine grid, we will construct a coarse grid by amalgamating (grouping) sets of cells into blocks B_j, $j = 1, \ldots, N$. Henceforth, γ_{ij} will denote the interface between two cells c_i and c_j and Γ_{ij} the interface between two blocks B_i and B_j. To represent amalgamated coarse grids, we will use a partition vector p with n elements, in which element p_i assumes the value j if cell c_i is member of block B_j. This definition of a coarse grid is very general and allows for blocks that potentially may be multiply connected. However, unless stated otherwise, we will

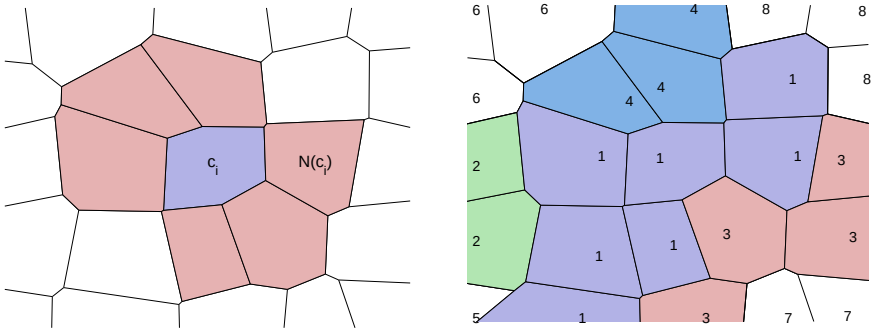

Figure 1—Example of a 2D unstructured grid; the left plot shows a cell and its nearest neighbors, whereas the right plot shows grouping of cells into blocks using a partition vector.

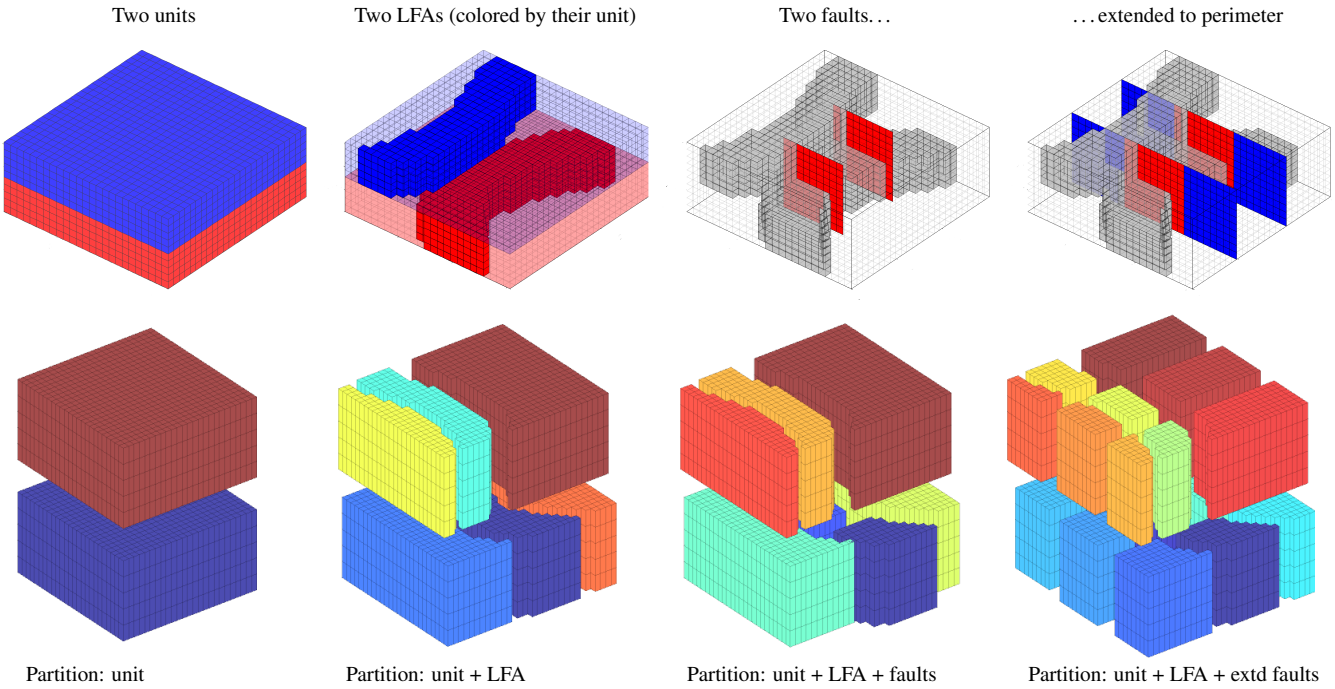

Figure 2—Illustration of the hierarchical coarsening approach for a geologic model with three features: unit, lithofacies assemblage (LFA), and faults.

henceforth tacitly assume that the blocks B_j are singly connected at all times.

To motivate the hierarchical coarsening, we will use a small conceptual model that has three main geologic features, as shown in Figure 2. The idea of the hierarchical coarsening algorithm is to define these features as partition vectors and apply them in sequence to define a sequence of models of increasing resolution. The lower part of the figures shows the result of first applying unit, then lithofacies assemblage (LFA), and finally the (extended) faults to partition the domain. This is just one choice, the features could have been applied in any other order, preferably starting with the most important feature, then adding the next most important, and so on. Examples of other potential features to use include environment of deposition (EOD), facies and zone indices, fluid and displacement regions (saturation, relperm, etc), layer identifiers, initialization regions, etc. Likewise, one can use flow information like time-of-flight to separate the reservoir into near-well, far-field, and intermediate regions or high-flow and low-flow regions. Ideally, coarsening workflows and software solutions should come with clear recommendations of which geologic or flow features to choose and in which order they should be applied. Unfortunately, we believe that this will vary from case to case and possibly also depend on the purpose for which the coarsened model is to be used. Our hierarchical approach has therefore been implemented as a set of basic operations that gives the user flexibility in using his/her understanding of the reservoir to make good hierarchical partitions.

The hierarchical approach outlined above will only provide relatively coarse blocks – the upscaling factors in Figure 2 range from 347 to 3125. To obtain coarse models with higher resolution, we must therefore apply some additional partitioning principle. There are a wide range of principles one can use to accurately account for the remaining heterogeneity that is not represented in

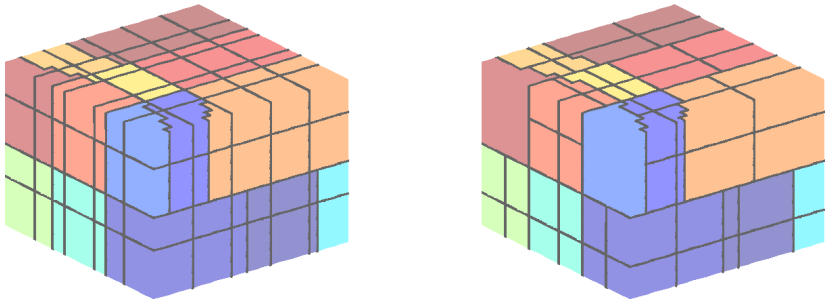

Figure 3—Illustration of two different refinements of a hierarchical partition shown in color, obtained by applying unit, LFA, and extended faults. To the left, all blocks are subdivided using a background 5×5×4 partition. To the right, the bounding box of each block that contains two hundred cells or more is subdivided uniformly into 2×2×2 subvolumes.

the geologic parameters used for the first hierarchical coarsening:

1. A simple way of generating a partition is to use a standard 'cookie-cutter' to subdivide regions of the reservoir into blocks of predefined shapes in physical or index space. This can be applied to all grid blocks within a region, or to all grid blocks whose (pore)volume exceed a user-prescribed threshold. For instance: for a corner-point grid, the 'cookie-cutter' could be a standard $n \times m \times k$ load-balanced index partition, or one could use the centroid of the grid cells to sample from a coarse Cartesian partition of the grid's bounding box. However, the user could also in principle prescribe coarse blocks with any shape and composition. If necessary, the process can be repeated using a series of partition vectors of increasing resolution.

2. A second alternative is to split the bounding box of each coarse block into a set of sufficiently small subvolumes. For a structured grid, this can be done in index space, but for other grids it is more natural to partition in physical space.

3. Another possibility is to use METIS (Karypis and Kumar 1998), or a similar graph-partitioning algorithm, configured with the transmissibilities of the fine-scale discretization as weights for the edge-cut minimization algorithm. In our experience, this is a simple and robust method for generating a coarse partition with a given number of blocks that provide a priori adaption to local features.

Figure 3 illustrates the first two approaches for our simple model. The advantage of using a regular background partition to subdivide blocks is that one ends up with a coarse partition that has matching faces, whereas the disadvantage is that one easily gets small blocks where the geologic features cut through the background partition. Small blocks can be avoided if we instead split the bounding boxes of the blocks independently, as shown to the right in Figure 3. The disadvantage of this approach is that it is difficult to split in a way that minimizes the number of non-matching coarse faces.

Non-uniform Coarsening by Amalgamation Algorithms

If the permeability has strongly channelized features that delineate high- and low-flow zones on a smaller scale than the geologic features used in the multilevel hierarchy, better accuracy may sometimes be achieved by adapting the partition to features in the permeability field or to the flow field, if this is available. In this section, we outline a general family of heuristic algorithms that produce coarse grids by amalgamating (grouping) cells from the underlying fine grid. The algorithms all rely on cell-wise indicators to set the local grid resolution and define how the cells should be grouped together to adapt to heterogeneity or flow patterns. In an early approach in this direction, Aarnes et al. (2007) suggested to group cells according to the magnitude of the velocity (or flux) field, i.e., so that each coarse block consists of a collection of cells having approximately the same magnitude of flow. To achieve such a partitioning, the authors proposed an *ad hoc* method consisting of four steps as illustrated in Figure 4. Later research by Hauge et al. (2012) and Hauge (2010) has shown that the original nonuniform coarsening (NUC) method is just a special case of a much larger family of amalgamation methods. In general, the direction in which the local amalgamation proceeds can be steered by two different types of rules: a neighbor definition that gives the *admissible* directions and a set of indicator functions that define the *feasible* directions. In addition, one needs criteria to determine how far the amalgamation should proceed, i.e., to determine the size of each grid block.

The admissible directions are basically given by the grid topology, but can be restricted (or extended) using geometric information or cell and face constraints; this will be discussed in more detail below. Feasible directions are defined by the indicator functions and a set of coarsening principles that work together with the indicators:

- Minimize the accumulative indicator value or the variation in indicator values inside each coarse block

- Design indicator functions so that good grids can be created by equilibrating indicator values over grid blocks. Example: using velocity magnitude as indicator will ensure high resolution in high-flow zones and lower resolution in low-flow zones.

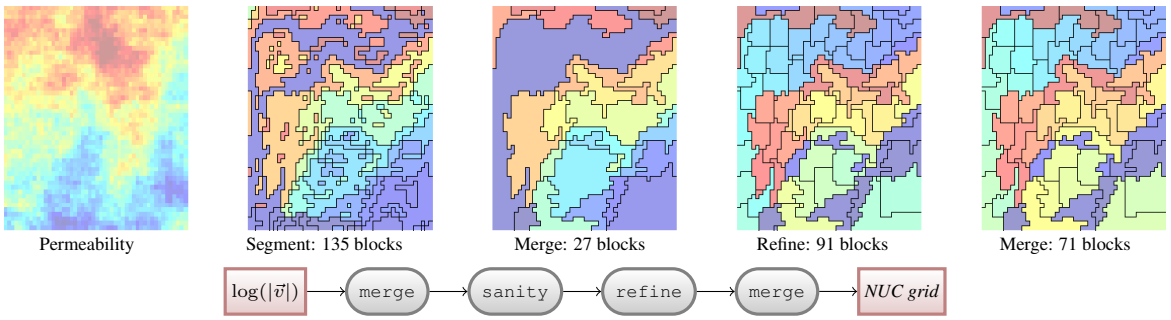

Figure 4—Example of an amalgamation algorithm, the nonuniform coarsening method (Aarnes et al. 2007), applied to a 50×50×1 excerpt from the SPE10 benchmark with a quarter five-spot well pattern. The merge steps use a volume indicator to merge small blocks, whereas the refinement step splits blocks with flow indicator exceeding an upper limit.

- Try to make the blocks as regular as possible by minimizing aspect ratio, irregularity, etc.
- Keep block sizes within prescribed upper and lower bounds.

The first principle tries to make the blocks as homogeneous as possible, e.g., by separating cells with high and low permeabilities in different blocks; forming blocks that consist of the same facies, stone type, or saturation region number; separating cells with low and high flow in different blocks; separating cells with low and high travel times in different blocks; and so on. The second principle can be used to increase the resolution in areas of particular interest, like high-flow regions or regions with high sensitivity indicators. This principle is inspired by a posteriori error analysis, in which one adapts the grid to equilibrate the error contribution from each cell. The last two principles seem natural from a standard discretization point-of-view.

Each algorithm can be described as a set of sources that each creates a partition vector and a set of filters that each operates on one or more partition vectors to create new partition vectors. Examples of such operations include computing the intersection of two or more partitions, performing sanity checks and repairs to ensure connected partitions, as well as modifying an existing partition by merging blocks that are too small with neighbors having similar indicator values or splitting blocks whose indicator values exceed some accumulative or variation criterion. For the splitting, we can use any of the subdivision methods discussed in the previous section. However, for the NUC class of algorithms it is more common to use a greedy algorithm that starts from a cell on the perimeter and gradually grows a new block by adding neighboring cells from within the existing block.

Indicator functions These have two main uses in our algorithmic framework: (i) to identify feasible amalgamation directions in the partition primitives and the refinement step, and (ii) to determine locally the correct size of the blocks during merging and refinement. In most cases, we will therefore operate with two types of indicators: an additive volume indicator (bulk or pore volume), and a 'flow' indicator that tries to measure features that will affect the flow. Our primary example is time-of-flight τ, which measures the time it takes a passive fluid particle to travel from the nearest fluid source to a given point in the reservoir, or alternatively the time it takes from a given point to the nearest fluid sink. Isocontours of τ define natural time-lines in the reservoir and time-of-flight is a natural ingredient when forming flow-based indicator functions, since it is a global quantity that depends on the flow paths and not only local heterogeneity. We compute time-of-flight by solving $\pm\vec{v}\cdot\nabla\tau^{\pm}=\phi$, where $\tau^{+}=0$ at fluid sources and $\tau^{-}=0$ at fluid sinks. Using $\min(\tau^{+},\tau^{-})$, we can easily distinguish near-well from far-field regions, and using the residence time $\tau^{+}+\tau^{-}$, we can distinguish high and low-flow zones.

Time-of-flight is not available by default in most flow simulators. Aarnes et al. (2007) therefore suggested to use the logarithm of the velocity (i.e., either $\log(|\vec{v}|)$ or $\log(|\vec{u}|)$) to get a local indicator that is based on a quantity that is available in most flow solvers. Several authors have also used vorticity, which is defined as the curl of the velocity field and tends to smooth velocity variations, and hence is a better estimator for grid density than velocity for smoothly varying heterogeneity. Other alternatives would be to use permeability or functions thereof, sensitivities derived from adjoint simulations, etc. Instead of using cell-based quantities, one can also face-based quantities like flux magnitude, transmissibilities, etc.

Admissible directions How the admissible directions is defined affect the intersection, merging, and refinement primitives and has a strong impact on the shape of the coarse blocks and the topology of the partition. The intersection primitive uses admissible neighbors to ensure that each block in the partition vector is singly connected. The merging primitive searches through admissible neighbors to find blocks that can be merged. Likewise, the greedy algorithm used to split blocks into multiple parts needs to know all admissible neighbors when growing a new block. How to define the neighborhood of a cell, is therefore one of the important algorithmic choices in our framework.

Low-order finite-volume methods used in reservoir simulation rely on face neighbors only, which we will refer to as level-one neighbors. We can also define level-two neighbors as those cells that share an edge and level-three neighbors as those who share a vertex. (In 2D, we obviously only have level-one and level-two neighbors.) Figure 6 shows the difference between

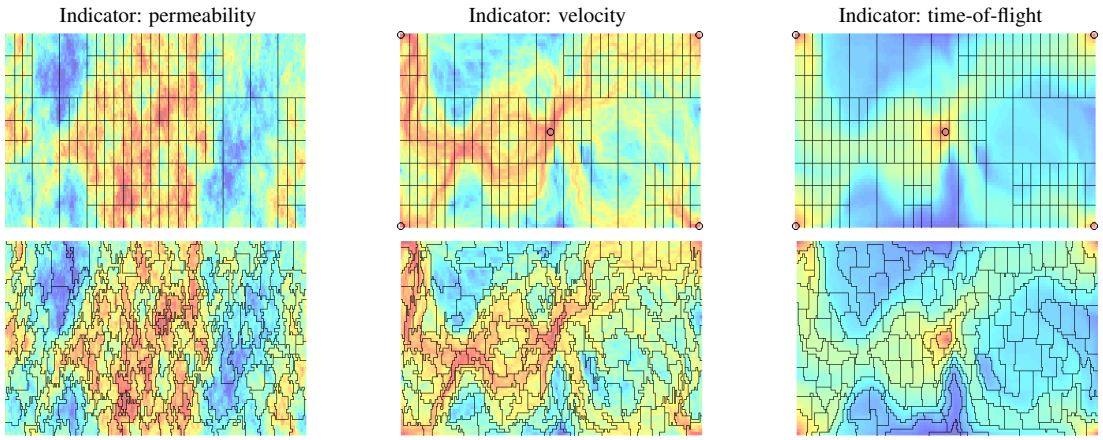

Figure 5—Illustration of the effect of different flow indicators. In the upper plot, we start by a uniform 3×11 partition of a single layer of the SPE10 model and then add an extra 3×3 refinement in all blocks where the flow indicator exceeds a prescribed limit. The lower plot shows NUC grids.

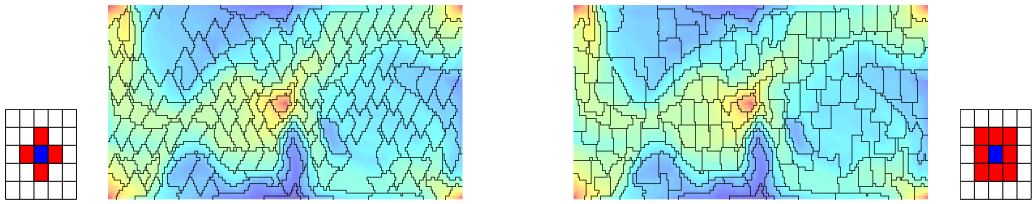

Figure 6—The original NUC algorithm with time-of-flight indicator for two different definitions of admissible neighbors.

using 5-point and 9-point connections in the refinement step of the original NUC algorithm from Figure 4. For triangular and tetrahedral grids, the use of level-one neighbors will typically lead to quite irregular interfaces between coarse blocks, and more regular block faces can be obtained if one extends the admissible neighbors to include cells that share faces with two or more face neighbors. However, algorithms can also use other definitions of admissible neighbors, e.g., only vertical or horizontal neighbors in stratigraphic grids. This type of restriction can be used to get coarsening methods similar to those discussed by King (2007) for merging vertical layers. For the refinement step, we could also use geometry information to define admissible neighbors e.g., as all cells whose centroids are within a certain distance from the given cell. Notice, however, that uncritical use of geometric neighborhoods may lead highly irregular sawtooth faces, blocks not respecting geologic layers, etc.

One purpose of the multilevel, hierarchical approach outlined briefly in the previous section is to pick geologic properties that the partitions should adapt to. This will in turn introduce hard constraints on the admissible directions when amalgamating cells into blocks. These constraints may come in the form of extra *a priori* (or background) partitions that should be preserved, e.g., so that a new block does not grow across borders between different EODs, LFAs, facies region, etc. To preserve such cell-based constraints throughout merging and refinement operations, we typically redefine the local topology to exclude connections between cells on opposite sides of the artificial internal boundaries imposed by the *a priori* partition vectors. Constraints can obviously also be imposed directly on the faces, e.g., so that growing blocks do not grow across faults, fractures represented as lower-dimensional objects in a discrete fracture network (DFN) model, etc. Face-based constraints do not correspond naturally to any partition vector and must therefore be introduced during the postprocessing phase of the intersection primitive by removing cell connections over the constrained faces. To preserve the constraints, the set of admissible neighbors should be manipulated by removing the same cell connections also during merging and refinement.

Example: Coarsening, Upscaling, and Workflow Application

Branets et al. (2015) recently presented a geometric-centric geologic modeling workflow aimed at accurately representing geologic features of potential importance to flow. In this workflow, the grid generation can work directly with the stratigraphic framework to build layering and truncate cell geometries by the horizons and faults from the structural framework without compromising on the representation of geologic complexity. In the following, we will illustrate the multilevel hierarchical coarsening on one particular high-resolution model generated by this workflow, see Figure 7. The model is discussed in detail in (Branets et al. 2015); here we only provide a brief review. The regional setting is retrograding from a transitional shoreface environment in the lower zone, through a shallow-water deltaic environment in the middle zone, and into a deep-water, channelized environment in the top zone. In the shoreface environment, boundaries between different EOD/facies may potentially act as barriers or baffles to flow and also separate contrasts in rock properties. Porosity distribution is smooth inside EOD regions. In the

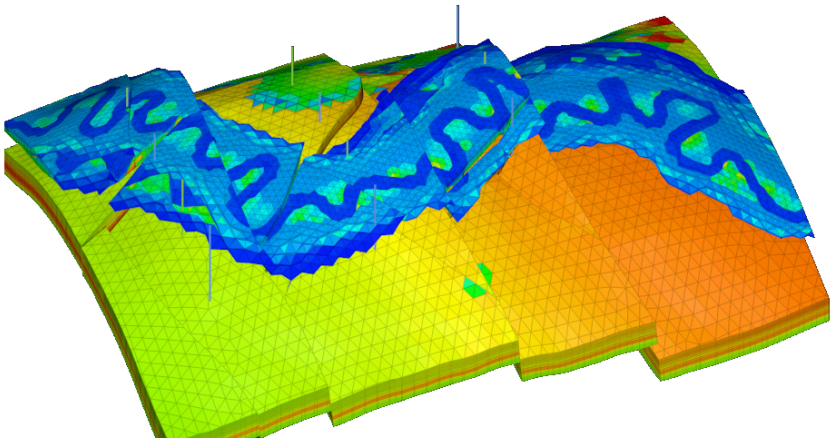

Figure 7—A geometry-centric geologic model represented as a layered prismatic (2.5D Voronoi) grid. In the plot, the 153 000 cells are colored by porosity value. We use a simulation setup with 11 wells (6 producers and 5 injectors).

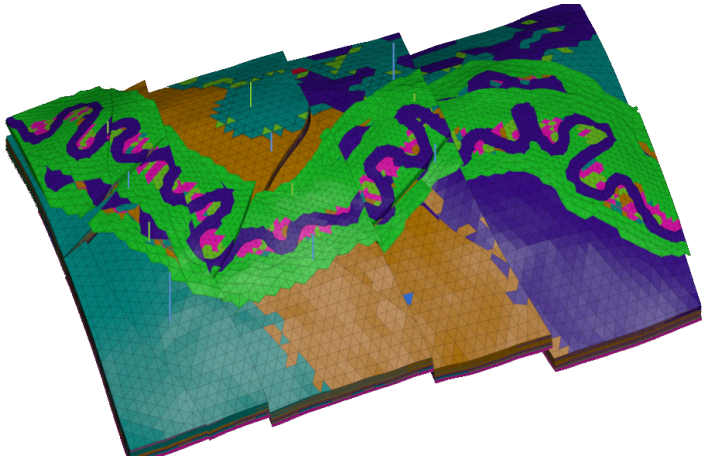

Figure 8—EOD regions preserved during the NUC construction using hierarchical coarsening.

deltaic environment, regions are defined by lobe and channel geometries. Mud drapes represent flow barriers and increase model compartmentalization, whereas erosive channels increase connectivity near the proximal part of the channel-lobe model. The deep-water environment consists of channel belts, each of which is formed by depositional point bars and erosional abandonment channel regions. Point bars are further subdivided by lateral accretion boundaries posing as potential flow baffles. Complex connectivity is created by channel belts eroding into each other. The final model is created by combining the three zones together and computing cell connections across faults and zone boundaries. The reservoir is produced by eleven wells, six producers and five injectors, all controlled by prescribed bottom-hole pressures.

Coarsening To develop a hierarchical coarsening, we first have to decide which geologic features to preserve when coarsening. Because the three zones are represented as distinct layers in the fine-scale grid, they are simple to preserve by modifying the definition of admissible neighbors so that all connections between zones are eliminated. This way, all blocks that encompass multiple zones will be split during the sanity check, and likewise, no new blocks are allowed to grow out of its originating zone during the merging or refinement steps. The second feature we will preserve are the EOD regions shown in Figure 8. Because of the many flow baffles and barriers, as well as differences in rock properties, the flow will be almost confined to a subdomain of the reservoir. To reduce computational cost while preserving as much accuracy as possible, it is therefore natural to construct a reduced model that maintains high spatial resolution in the high-flow zone but uses more aggressive coarsening in stagnant regions and regions with small flow. These regions are not easy to elucidate from geologic information only. Instead, we therefore solve a single-phase pressure problem to get a representative flow field, from which we can compute time-of-flight values τ^+ from injectors and τ^- to producers. We then use these time-of-flight values to subdivide the reservoir into different regions: the near-well region in which $\min(\tau^+, \tau^-)$ is below a prescribed value, the far-field region in which time-of-flight is large or cells are not in communication with any of the wells, and potentially also a transition region in between. In the near-well region, we essentially use the original NUC method (Aarnes et al. 2007) with time-of-flight as flow indicator. In the far-field region, we use the NUC

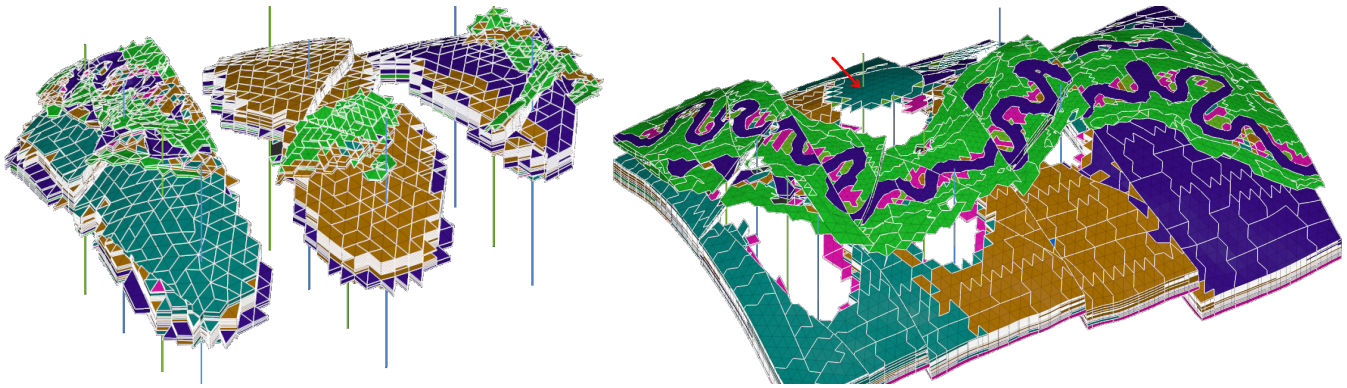

Figure 9—The left plot shows the near-well region defined using time-of-flight and coarsened to a smaller degree by the NUC algorithm with time-of-flight as flow indicator and pore volume as volume indicator. Blue are water injectors, green are producers. The right plot shows the far-field region, which was more aggressively coarsened by the NUC algorithm with a 'cookie-cutter' refinement step (using geometric IJK information along with flow indicators)

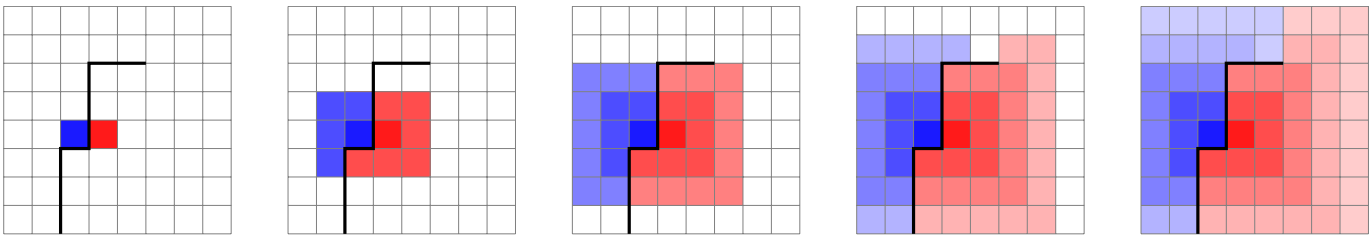

Figure 10—Greedy algorithm for splitting a block that is partially transversed by a (sealing) fault.

algorithm with one initial bin and a cookie-cutter algorithm based on pseudo-IJK information supplied by the grid instead of the greedy algorithm in the refinement step. Figure 9 shows the partitioning inside the near-well and far-field regions (in this case, we did not use a transition region). The careful reader may notice that one of the wells (indicated by a red arrow) penetrates a relative large block in the far-field region. The reason is that the well is not perforated in this layer of the reservoir. Because the near-well region is defined on basis of flow and not geometry, the cells inside this block are in the far-field region.

Last, but not least, we ensure that blocks do not cross faults. It is straightforward to split all blocks that are transversed by a fault surface by defining the admissible directions so that all cell connections are eliminated across faults. The challenge is to also split blocks that are only partially penetrated by a fault surface. Such blocks are not problematic if the fault is open and fully permeable. If the fault is sealing, however, these blocks may induce wrap-around effects that adversely affect the resolution of flow barriers and baffles if left unattended. In Figure 2, we showed a simple case where the fault planes followed the axial directions and it was easy to extend each of them to the perimeter of the reservoir. For a complex high-resolution model like the one considered in this section, such an approach would easily lead to intricate and expensive geometric processing requiring information that generally is not available from the fine grid model. Instead, we will use a relatively simple greedy algorithm, as illustrated in Figure 10. The algorithm starts by picking two (neighboring) cells on opposite sides of the fault that each is assigned to a new block. We then continue to grow the blocks outward one step at the time by adding all cell-neighbors to each of the two blocks; in the figure, we have used level-two neighbors. A new neighboring cell will not be added to the block if it shares a fault face with another cell that is already in the new block. If the original block contains multiple partial faults, the same process is repeated for each of the faults. This algorithm is admittedly crude, but seems to work well in practice. Figure 11 shows a zoom of the model around a set of intersecting faults. The resulting coarse grid with approximately 21 000 blocks is shown in Figure 12. Altogether, the non-uniform coarsening – which uses a combination of geo-hierarchy, modest flow-based coarsening in the near-well zone, aggressive cookie-cutter partitioning in the far-field zone, and fault processing – gives a very good representation of the stratigraphic and structural frameworks.

Upscaling A large variety of single-phase upscaling techniques exist that can be used to compute effective permeabilities and transmissibilities on the coarse grid (Durlofsky 2003, 2005; Christie 1996; Farmer 2002; Stern 2005). For our model-reduction purposes, well positions will mostly be fixed. Accurate upscaling can then be obtained if we use a flow-based method with specific global boundary conditions to compute effective transmissibilities and well indices (r-values). The main problem with such methods is that they tend to generate negative transmissibilities in many of the blocks in the presence of strong heterogeneities. To a certain extent, this tendency is reduced by using a hierarchical coarsening strategy that adapts to geologic features. In addition, sequential fallback options are used such as upscaling using a local pressure solve or simple arithmetic-harmonic averaging.

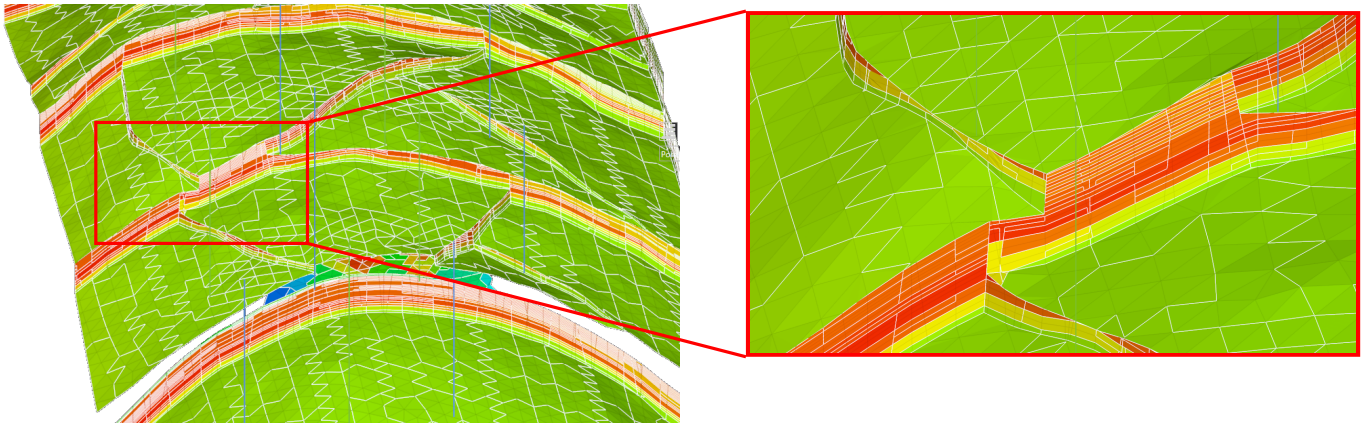

Figure 11—Bottom view (zommed out and zoomed in) to show intersecting faults and coarse grids honoring faults. Black lines are the underlying fine grid, and white lines are the coarse grid wireframe.

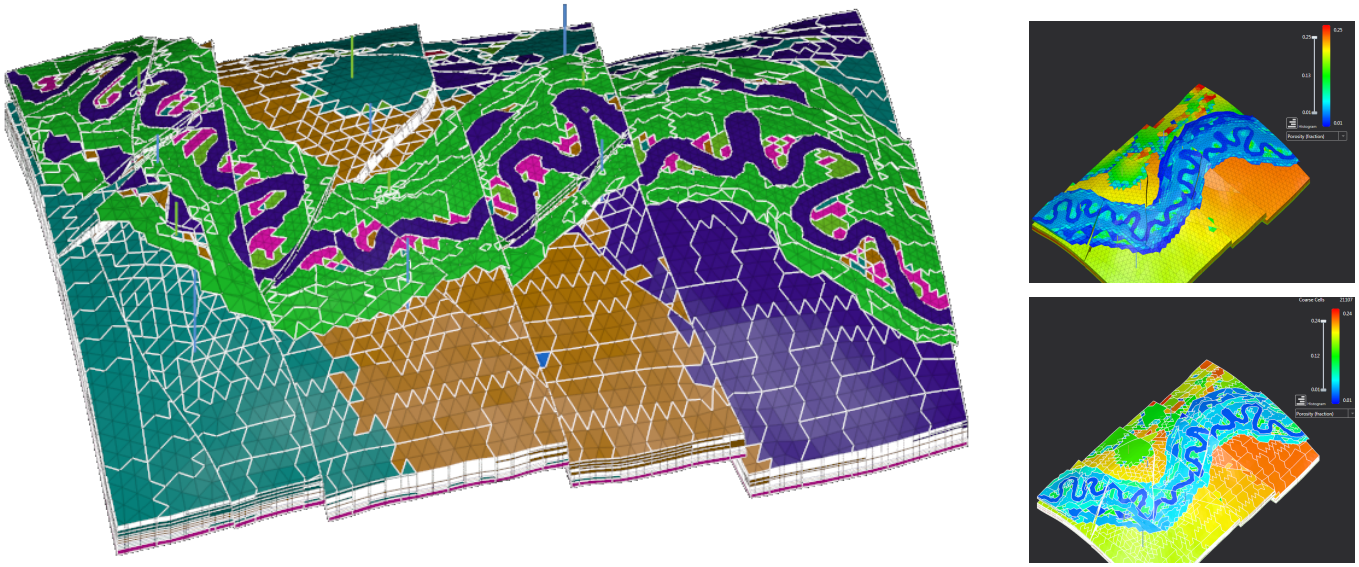

Figure 12—The non-uniform coarse grid. The left plot shows the overall partition into approximately 21 000 block, whereas the right column compares the porosity on the original grid (top) and on the coarse grid (bottom).

Altogether, this gives a robust method.

To test accuracy of the upscaling, we consider a black-oil reservoir, initialized with only oil and water (no gas cap). The relative permeability and capillary pressure curves used in the model corresponds to a mixed wettability rock. We use gravity equilibrium to initialize the model. The average oil saturation and reservoir pressure are 0.85 and 4000 psi, respectively. Injector and producer locations can (barely) be seen in Figures 7, 8, and 9. All the five injectors are water injectors, constrained at a high bottom-hole pressure of 5000 psi. The six producers are constrained at a low pressure of 3500 psi. In this example, we were interested in testing an oil-water system and hence a fluid with a low bubble point of 1500 psi is used to ensure no solution gas is evolved in the reservoir. Figure 13 shows the result of a fully coupled, implicit, three-phase simulation of this water-flooding scenario. After 7305 days of production, the discrepancy in cumulative production between the original high-resolution and our reduced-order model is 5.4% for oil, 5.0% for water, and 5.5% for gas. The mismatch in cumulative production provides a metric for screening out reduced order models but is not adequate to guarantee rank and trend accuracy. A reasonable indicator of trend accuracy could be similarity in the shape of production profile of high fidelity and reduced order models. For rank accuracy, a reasonable estimate could be the similarity in the ranking of well performance for high fidelity and reduced order models. In Figure 14, we test the rank accuracy by showing a well-by-well comparison of the production predicted by the original high-resolution model and our reduced-order model. Ideally, the dots should fall on the 45 degree line if the upscaled model was perfect. The distance between this line and the red dots, averaged over 90 day time interval for all of the 7305 days, is defined as rank error. The rank errors are 1.8% for oil, 6.3% for water, and 1.8% for gas, suggesting that the reduced order model has a good rank accuracy.

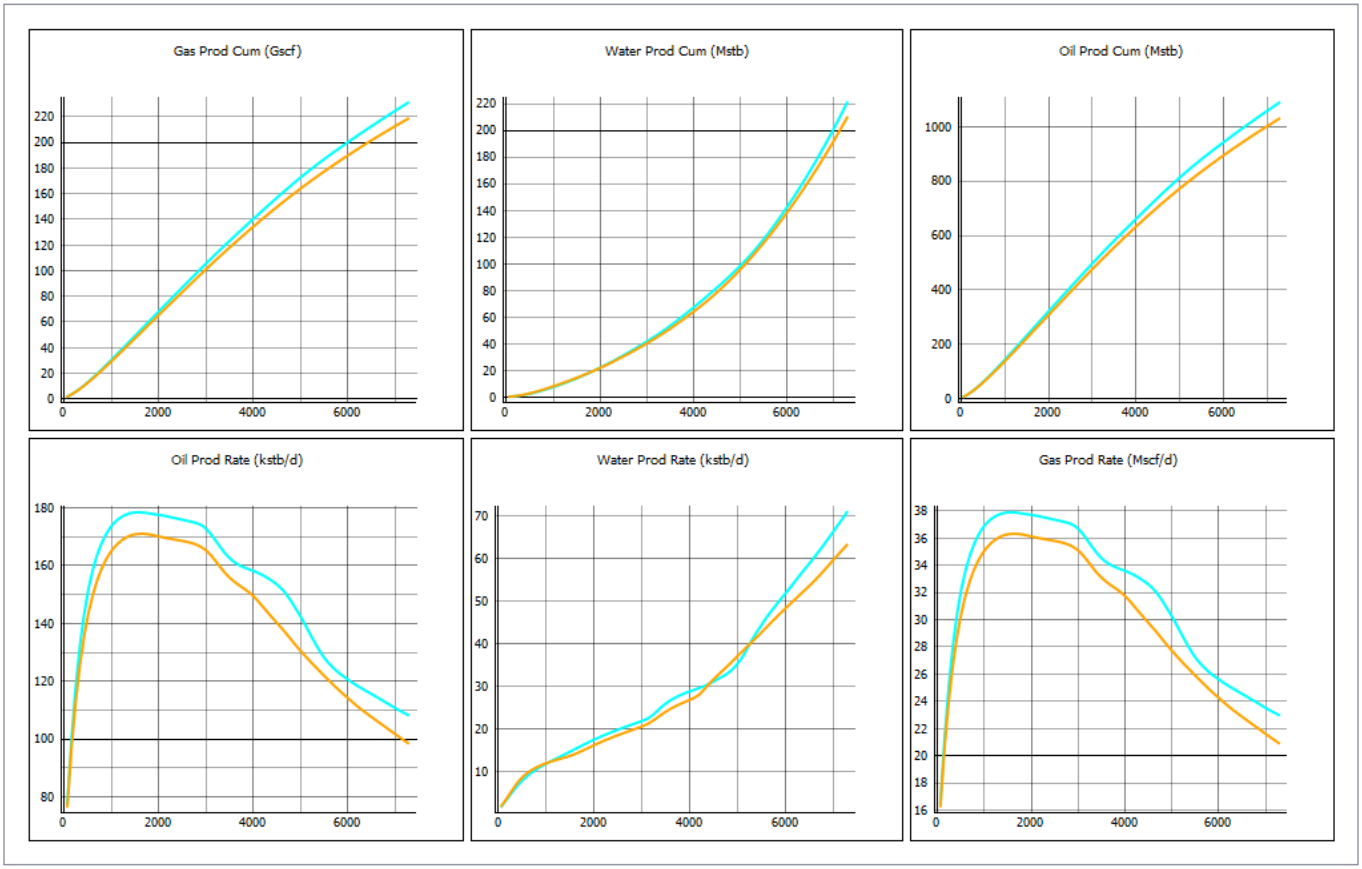

Figure 13—Total production curves for all eleven wells for a waterflooding scenario computed on the original high-resolution model with 153K cells (blue) and on the reduced-order NUC model with 21K cells (orange).

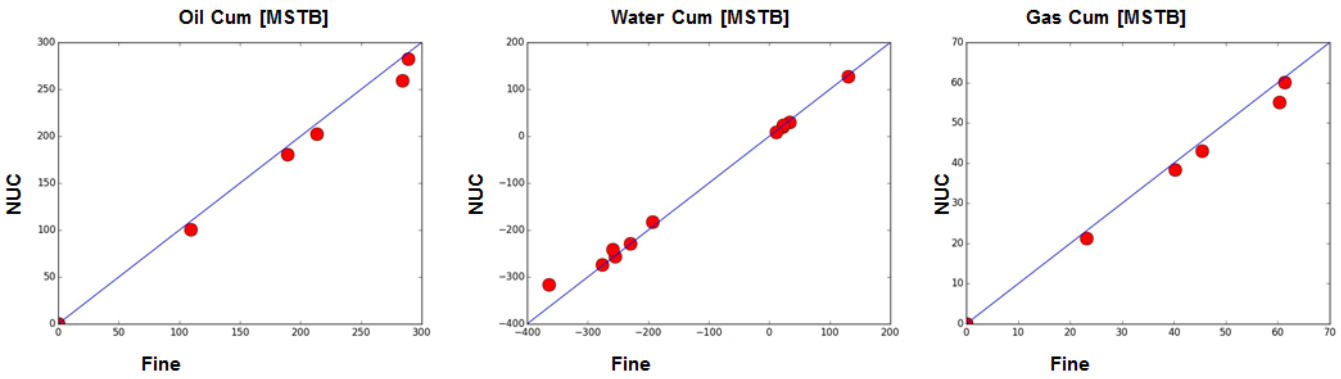

Figure 14—Well-by-well comparison of production at end time 7305 days. The figures show cross plot of simulation with fine grid versus the reduced-order NUC grid. Each red dot represents value for a well.

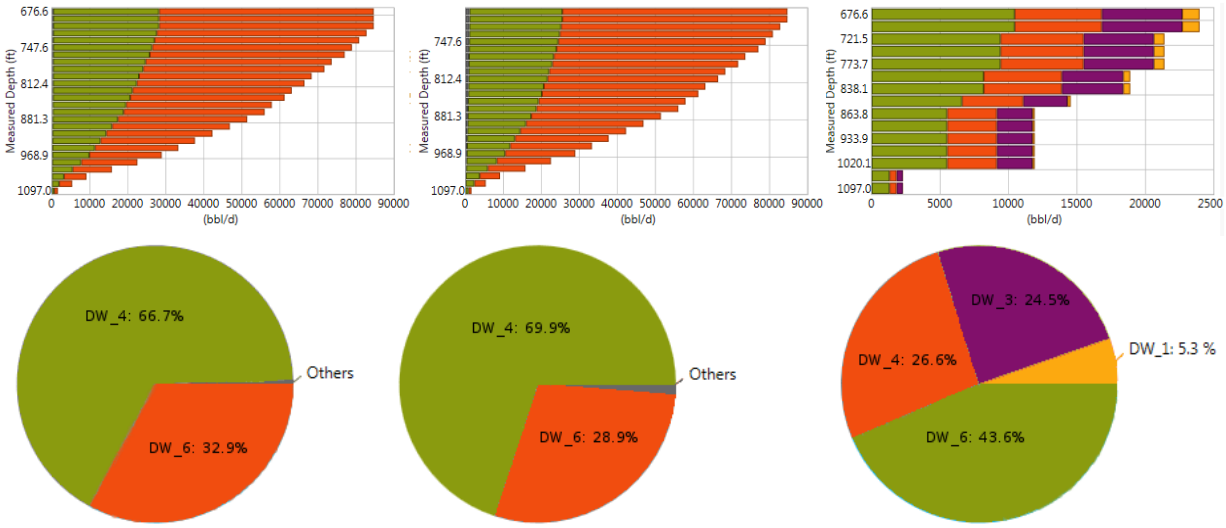

Figure 15—Illustration of how well-allocation factors can be used to verify coarsening and upscaling. The upper plot shows cumulative well-allocation factors from toe to heel, whereas the lower plots show overall well-allocation. From left to right: the fine-scale model, a coarse model with good match, and a coarse model with poor match.

Verification: flow diagnostics To be able to assess the quality of a given upscaling upfront without the need to perform extensive multiphase studies, we compute volumetric tracer partitions from the fine-scale flow field used for the specific upscaling. From this tracer partition, we compute the fraction of the inflow to a given producer that can be attributed to each of the injectors, and vice versa, the fraction of the outflow from a given injector that goes to each of the producers (Shahvali et al. 2012). These fractional fluxes are computed for each perforated cell, accumulated from toe to heel, and plotted as a function of distance from the well's toe to mimic a production log. The total well-allocation factor then equals the cumulative factor evaluated at the heel of the well. By computing the same quantities for the upscaled model and plotting the well-allocation factors on top of those from the original model (see Lie (2015; Chap. 10)), we get a visual flow diagnostics tool that simplifies the process of detecting and understanding potential mismatches in model predictions. Figure 15 shows one such flow diagnostics check for two different coarse models, one with good and one with poor approximation of the fine-scale model. The good model has the same well-pair communication as the fine model, but the allocation factors are slightly different. The poor model introduces well-pair communications that are not present in the fine model.

In our workflow, we have mainly used the mismatch in well-allocation factors predicted by a single-phase, slightly compressible flow simulation. This provides a visually intuitive quality check for individual wells and can be utilized for sectors or the reservoir as a whole as long as the number of wells is not too large. If you consider an area that only contains a few well pairs, it may be necessary to subdivide wells into two or more segments and look at segment pairs rather than well pairs to make the flow diagnostics predictive. To better understand mismatches, it is also straightforward to compute drainage or sweep regions for individual well completions so that these can be contrasted and compared for the original and upscaled model. For reservoirs with hundreds of wells, this type of visual flow diagnostics is no longer informative, and more research is needed to develop more automated methods for measuring quality of the upscaling.

Workflow application The reduced models retain sufficient predictive power to confidently accelerate computationally intensive tasks like history matching, optimization and uncertainty quantification. As a last example, we will utilize the reduced-order model to perform a sensitivity analysis. To this end, 16 different cases were created using permutations of the following parameters: transmissibility multiplier (10^{-4} factor) as a baffle for Faults 1 to 3, horizontal permeability multiplier throughout the reservoir (0.7 low side, 1.3 high side), and porosity multiplier throughout the reservoir (0.7 low side, 1.3 high side). Figure 16 shows that the NUC model predicts similar trends as the original fine model, while giving a 30-times computational speedup. This demonstrates that a well-constructed NUC model can help explore uncertain space rapidly.

Conclusions

In this paper, we have described a general hierarchical framework in which a set of geologic features are used to gradually subdivide the reservoir into smaller and smaller volumes. To create specific coarsening algorithms, one must identify a set of cell-based or face-based properties and specify in which order these are to be applied. If wanted, the hierarchical approach can be combined with flow-adapted griding. In many cases, the default choice is to use some kind of regular partition of the fine-scale model to form a coarse grid. By extending such regular partitions to adapt to geologic features like faults, boundaries of flow units, environments of deposition, lithofacies assemblage, and so on, our hope is to introduce a larger degree of robustness and

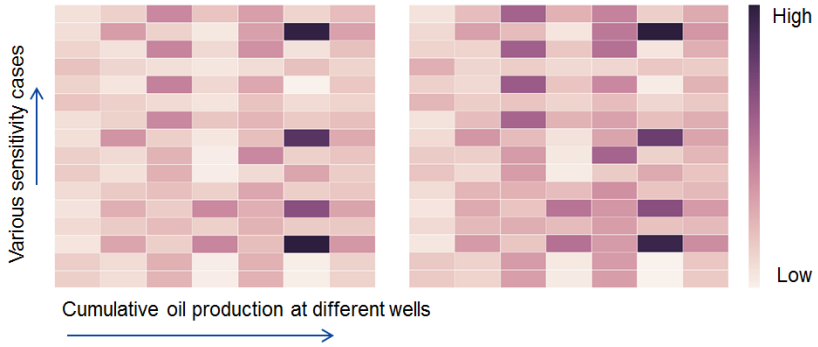

Figure 16—Sensitivity analysis computed by original fine-grid model (left) and the reduced-order NUC model (right).

make the upscaling less sensitive to user-prescribed coarsening parameters such as the number of blocks in each spatial direction. For the upscaling, we have used a global flow-based method with specific boundary conditions, with a two-level fallback strategy that first uses a local flow-based method and then a simple averaging procedure to avoid negative transmissibilities.

Early versions of these ideas (hierarchical coarsening and flow diagnostics) have previously been tested independently by SINTEF on a large variety of idealized and synthetic models, but have so far not been systematically reported in published literature. Herein, we report our experience based on implementing the same methods in a full-fledged simulator environment and validating it on a large variety of geologic models of real assets. The summary of our experiments is that the irregularity of adapted grids may in some cases introduce a somewhat higher upscaling error, but in most cases the adaption will lead to significant improvement in the upscaling. Generally, the variation in upscaling errors is smaller when using adaption than when just using regular partitions.

Based on our experience, we suggest the following guidelines for creating good coarse grids:

1. Keep the grid as simple as possible: use regular partitions for mild heterogeneities and cases where saturation/concentration profiles are expected to not contain a lot of fingering.

2. Adapt coarse blocks to avoid upscaling large permeability contrasts, constrain blocks to contiguous saturation regions to avoid upscaling k_r and p_c curves, etc. Adapting to geologic features may prove to be more important when upscaling multiphase flow properties: if a block consists of a single stone type, one may not have to upscale the relative permeability and capillary pressure curves for this block. However, for this to be useful, the stone types must form contiguous regions and be devoid of too much small-scale speckle.

3. Use basic geologic features when available. Example: using facies is likely more robust than segmenting permeability into bins. In picking the geology parameters, you should try to determine the features that separate high-flow from low-flow regions (flow units, lithographic assemblage, deposition environments, facies, etc), what are the features that determine large-scale flow directions (channels, barriers, etc), and so on. Once identified as cell-based or face-based quantities in the fine grid, these features can easily be imposed in the coarsening procedure using variants of the hierarchical procedures discussed above.

4. When using faults to subdivide coarse blocks, try to extend the faults that die out in the middle of the block to the block perimeter to avoid creating blocks with wrap-around that potentially could lead to poor resolution of barrier effects. For simple models, one can extend each fault plane to the reservoir perimeter and use a 'soft fault' concept to avoid refining too large regions. Basically, this concept means that blocks that are traversed by a fault and its extension are split, whereas blocks that are only traversed by a fault extension remain intact. For more complex models, we proposed a simple greedy algorithm that starts to grow new blocks on opposite sides of each fault plane.

5. In many cases, the geologic parameters will not provide a coarse grid of sufficient resolution and additional subdivision principles may be needed. You should then look at the heterogeneity inside each of the 'compartments' that have been identified by your geologic features to pick additional 'background partitions' for subdividing each compartment. If the petrophysical parameters are mildly heterogeneous inside a compartment, you should use a regular subdivision. If a compartment has channelized features, you may consider introducing flow-based subdivision.

6. If you want to include some kind of flow adaption, use simple indicator functions that have an intuitive interpretation like velocity magnitude, time-of-flight/travel time, etc.

7. Avoid grids with large differences in block volumes (and the number of fine-scale cells that are contained in each block) in the parts of the reservoir where most of the flow takes place. In the far field, however, it is usually safe to use more aggressive coarsening.

8. Try to design your partitioning algorithm so that you reduce the need for black-box merging algorithms.

9. Check the quality of the coarse grid and the subsequent upscaling by computing simple flow diagnostics (Lie 2015).

Acknowledgments

The authors thank ExxonMobil Upstream Research Company for permission to publish this paper.

References

Aarnes, J. E., Hauge, V. L., and Efendiev, Y. 2007. Coarsening of three-dimensional structured and unstructured grids for subsurface flow. *Adv. Water Resour.*, 30(11):2177–2193. doi: 10.1016/j.advwatres.2007.04.007.

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2006. A hierarchical multiscale method for two-phase flow based upon mixed finite elements and nonuniform coarse grids. *Multiscale Model. Simul.*, 5(2):337–363. doi: 10.1137/050634566.

Aarnes, J. E., Krogstad, S., and Lie, K.-A. 2008. Multiscale mixed/mimetic methods on corner-point grids. *Comput. Geosci.*, 12(3):297–315. doi: 10.1007/s10596-007-9072-8.

Alpak, F. O., Pal, M., and Lie, K.-A. 2012. A multiscale method for modeling flow in stratigraphically complex reservoirs. *SPE J.*, 17(4):1056–1070. doi: 10.2118/140403-PA.

Ashjari, M. A., Firoozabadi, B., and Mahani, H. 2008. Using vorticity as an indicator for the generation of optimal coarse grid distribution. *Transp. Porous Media*, 75(2):167 – 201. doi: 10.1007/s11242-008-9217-9.

Branets, L., Kubyak, V., Kartasheva, E., Shmyrov, L., and Kandybor, D. 2015. Capturing geologic complexity in simulation grid. In *SPE Reservoir Simulation Symposium, Houston, Texas, 23–25 February*, number SPE 173270-MS.

Branets, L. V., Ghai, S. S., Lyons, S. L., and Wu, X.-H. 2009. Review article: Challenges and technologies in reservoir modeling. *Commun. Comput. Phys.*, 6(1):1–23.

Castellini, A., Edwards, M. G., and Durlofsky, L. J. 2000. Flow based modules for grid generation in two and three dimensions. In *Proceedings of 7th European Conference on the Mathematics of Oil Recovery, Baveno, Italy, Sept. 5-8*.

Christie, M. A. 1996. Upscaling for reservoir simulation. *J. Pet. Tech.*, 48(11):1004–1010. doi: 10.2118/37324-MS.

Cirpka, O. A., Frind, E. O., and Helmig, R. 1999. Streamline-oriented grid generation for transport modelling in two-dimensional domains including wells. *Adv. Water Resour.*, 22(7):697 – 710. doi: 10.1016/S0309-1708(98)00050-5.

Durlofsky, L. J. 2003. Upscaling of geocellular models for reservoir flow simulation: A review of recent progress. Presented at 7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany, June 23–27, 2003.

Durlofsky, L. J. 2005. Upscaling and gridding of fine scale geological models for flow simulation. Presented at 8th International Forum on Reservoir Simulation Iles Borromees, Stresa, Italy, June 20–24, 2005.

Durlofsky, L. J., Jones, R. C., and Milliken, W. J. 1997. A nonuniform coarsening approach for the scale-up of displacement processes in heterogeneous porous media. *Adv. Water Resour.*, 20(5-6):335 – 347. doi: 10.1016/S0309-1708(96)00053-X.

Evazi, M. and Mahani, H. 2010a. Generation of Voronoi grid based on vorticity for coarse-scale modeling of flow in heterogeneous formations. *Transp. Porous Media*. doi: 10.1007/s11242-009-9458-2.

Evazi, M. and Mahani, H. 2010b. Unstructured-coarse-grid generation using background-grid approach. *SPE J.*, 15(2):326–340. SPE-120170-PA.

Farmer, C. L. 2002. Upscaling: a review. *Int. J. Numer. Meth. Fluids*, 40(1–2):63–78. doi: 10.1002/fld.267.

Garcia, M. H., Journel, A. G., and Aziz, K. 1992. Automatic grid generation for modeling reservoir heterogeneities. *SPE Reservoir Eng.*, 7(6):278–284.

Hauge, V. L. 2010. *Multiscale Methods and Flow-based Gridding for Flow and Transport in Porous Media*. PhD thesis, Norwegian University of Science and Technology.

Hauge, V. L., Lie, K.-A., and Natvig, J. R. 2010. Flow-based grid coarsening for transport simulations. In *Proceedings of ECMOR XII–12th European Conference on the Mathematics of Oil Recovery*, Oxford, UK. EAGE.

Hauge, V. L., Lie, K.-A., and Natvig, J. R. 2012. Flow-based coarsening for multiscale simulation of transport in porous media. *Comput. Geosci.*, 16(2):391–408. doi: 10.1007/s10596-011-9230-x.

He, C. and Durlofsky, L. J. 2006. Structured flow-based gridding and upscaling for modeling subsurface flow. *Adv. Water Resour.*, 29(12):1876 – 1892. doi: 10.1016/j.advwatres.2005.12.012.

Karypis, G. and Kumar, V. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comp.*, 20(1):359–392. doi: 10.1137/S1064827595287997.

King, M. J. 2007. Recent advances in upgridding. *Oil & Gas Science and Technology - Rev. IFP*, 62(2):195 – 205. doi: 10.2516/ogst:2007017.

King, M. J., Burn, K. S., Muralidharan, P. W. V., Alvarado, F., Ma, X., and Datta-Gupta, A. 2006. Optimal coarsening of 3D reservoir models for flow simulation. *SPE Reserv. Eval. Eng.*, 9(4):317 – 334. doi: 10.2118/95759-PA.

Lie, K.-A. 2015. *An Introduction to Reservoir Simulation Using MATLAB: User guide for the Matlab Reservoir Simulation Toolbox (MRST)*. SINTEF ICT, `http://www.sintef.no/Projectweb/MRST/publications`, 2nd edition.

Mahani, H. and Muggeridge, A. H. 2005. Improved coarse grid generation using vorticity. In *14th Europec Biennial Conference, Madrid, Spain, 13–16 June*, number SPE 94319-MS.

MRST 2016. The MATLAB Reservoir Simulation Toolbox, version 2016a. `http://www.sintef.no/MRST/`.

Natvig, J. R., Lie, K.-A., Krogstad, S., Yang, Y., and Wu, X.-H. 2012. Grid adaption for upscaling and multiscale methods. In *Proceedings of ECMOR XIII–13th European Conference on the Mathematics of Oil Recovery*, Biarritz, France. EAGE.

Natvig, J. R., Skaflestad, B., Bratvedt, F., Bratvedt, K., Lie, K.-A., Laptev, V., and Khataniar, S. K. 2011. Multiscale mimetic solvers for efficient streamline simulation of fractured reservoirs. *SPE J.*, 16(4):880–888. doi: 10.2018/119132-PA.

Prevost, M., Lepage, F., Durlofsky, L. J., and Mallet, J.-L. 2005. Unstructured 3D gridding and upscaling for coarse modelling of geometrically complex reservoirs. *Petrol. Geosci.*, 11(4):339–345. doi: 10.1144/1354-079304-657.

Shahvali, M., Mallison, B., Wei, K., and Gross, H. 2012. An alternative to streamlines for flow diagnostics on structured and unstructured grids. *SPE J.*, 17(3):768–778. doi: 10.2118/146446-PA.

Stern, D. 2005. Practical aspects of scaleup of simulation models. *J. Petrol. Tech.*, 57(09):74–81.

Verma, S. and Aziz, K. 1997. A control volume scheme for flexible grids in reservoir simulation. In *SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June*, number SPE37999. doi: 10.2118/37999-MS.

Wen, X. H., Durlofsky, L. J., and Edwards, M. G. 2003. Upscaling of channel systems in two dimensions using flow-based grids. *Transp. Porous Media*, 51(3):343 – 366. doi: 10.1023/A:1022318926559.