ACCURATE COARSE RESERVOIR MODELING USING UNSTRUCTURED GRIDS, FLOW-BASED UPSCALING AND STREAMLINE SIMULATION

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Mathieu Prévost

December 2003

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. Louis J. Durlofsky
(Principal Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. Khalid Aziz

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Dr. Mohammad Karimi-Fard

Approved for the University Committee on Graduate Studies:

Abstract

Geological characterizations typically produce detailed geocellular models that display high degrees of heterogeneity and geological complexity. These models are in general too detailed for direct use in reservoir simulators. For this reason, some type of coarsening procedure must be applied to obtain the flow model from the detailed geological model. This work addresses the generation of accurate coarse models that are able to capture complex reservoir geometries and at the same time preserve the basic flow response of the detailed model. This requires the use of unstructured gridding techniques and the application of specialized averaging or upscaling methods to capture subscale permeability heterogeneity. A computationally efficient flow-based diagnostic, capable of assessing the quality of the coarse model, is also an important component of the overall methodology.

In this work, new capabilities in all of these areas are presented. The techniques developed are suitable for use with a multipoint unstructured finite volume numerical scheme. The finite volume grid is formed as the dual of the underlying tetrahedral (in three dimensions) or triangular (in two dimensions) primal grid. A streamline simulation technique for the efficient modeling of flow and transport on unstructured grids is presented. This tool can be used to provide the flow-based assessment of the coarse model. The streamline method applied here considers unit mobility (tracer) displacements. The key to accurate

streamline simulation on unstructured grids entails the determination of an appropriate velocity interpolant. The technique presented here consists of a local postprocessing of the numerically calculated fluxes, leading to a consistent and flux-continuous piecewise constant representation for the velocity defined on a subgrid of tetrahedra or hexahedra. The resulting tracing acts to minimize the nonphysical swirling of the streamlines and is shown to provide accurate results for several reference problems.

The effects of fine scale heterogeneity in the coarse model are captured through the use of permeability or transmissibility upscaling. The permeability upscaling involves the local calculation of an equivalent permeability tensor for each coarse scale control volume. The transmissibility upscaling requires the calculation of the flow over the region corresponding to two coarse simulation cells and the subsequent determination of the flow through the interface between the two cells. The permeability upscaling is appropriate for use with multipoint flux approximations while the transmissibility upscaling is applicable within the context of two-point flux approximations. For the problems considered here, the transmissibility upscaling appears to be the superior technique, though more investigation of this issue is required.

The unstructured grid can be adapted to both geological features as well as to specific flow problems. Flow adaptation is accomplished through the use of single phase fine scale flow solutions, which are used to generate target grid resolution maps. Grids can be adapted to contain fine cells in regions of high flow or in regions leading to the early breakthrough of injected fluid. The unstructured grid generation techniques have been implemented in a Gocad-based software platform that allows for the construction of problem-specific adapted grids. The accuracy of the overall methodology is demonstrated through application to large synthetic reservoirs. Results are presented in terms of global flow rates and

water cut. It is shown that the new techniques are able to provide coarse scale descriptions of reasonable accuracy even for very high levels of coarsening. It is further demonstrated that the use of optimal cell aspect ratios and/or flow information can act to improve the accuracy of the coarse scale model considerably.

DEDICATED WITH LOVE TO

SEVERINE AND CONSTANTIN

Acknowledgements

Acknowledging the persons who drove/pushed me through this personal achievement appears not to be a trivial task.

Words are missing to express my immense gratitude towards my advisor, Prof. Louis J. Durlofsky. His guidance has been critical at times where each piece of our puzzle seemed to be failing in turn. Lou managed to strike the delicate balance between stimulating creativity and stopping fruitless stubbornness. Only now do I begin to grasp the purpose of this exercise. I realize that parts must have been frustrating, but I hope I have shown that I really did learn a lot.

I would also like to thank the members of my reading committee, Prof. Khalid Aziz and Dr. Mohammad Karimi-Fard, for their interest and input to this work.

Two months spent in the LIAD office in Nancy, France gave me a formidable push toward progress on the 3D developments. I thank Prof. Jean-Laurent Mallet for allowing me to interact with his team of "Gocadians" and for giving me access to Gocad technology. A big 'thank you' also to Dr. François Lepage and Dr. Guillaume Caumont for their help in using and developing Gocad plugins. The financial support of the SUPRI-B affiliates and the US Department of Energy is also gratefully acknowledged.

I thank my friends from Stanford and beyond who, for many, have been supportive beyond reason: Burak, Per, Tudor, Alex, H^3, Herve and Darryl. Moments of joy, hope and

occasional bits of sanity were provided by my wife Séverine. You kept us afloat in both stormy and fair weather, you are the strength behind each word of this manuscript, please accept the expression of my everlasting love and appreciation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Reservoir simulation is an essential tool for the management of oil fields. It can also be used to quantify the effect of uncertainty in reservoir description. Reservoir engineers are faced with the challenge of evaluating multiple flow scenarios on possibly multiple realizations (or views) of the reservoir. With geological models exceeding current computer capabilities for simulation purposes, significant coarsening of the reservoir description is generally needed to obtain and process the flow simulation results in the required time. To attain such coarsening, flexible (unstructured) grids provide an attractive solution to enable the accurate and efficient modeling of the reservoir geometry and heterogeneity.

The work presented here aims at providing semi-automated tools to generate coarse reservoir models. Highly detailed reservoir descriptions are assumed to be available and we consider the task of constructing, in an efficient manner, coarse grid models that "optimize" the number and distribution of the grid cells needed to accurately represent the detailed model. In this approach, the coarse models constructed are intended to reproduce flow simulation results obtained on the reference model under a set of particular boundary conditions. Obtaining a match for displacement problems while considering multiple

flow scenarios is also possible using the methodology presented. Streamline simulation is the preferred tool to evaluate the match between coarse and fine models and also to obtain valuable spatial flow related information. Various types of flow information are extracted from these simulations and integrated in the gridding process.

An important contribution of this work is the extension of the streamline method to three-dimensional unstructured grids. We also implement techniques for the modeling of complex geological features of the reservoir and develop flow-based 3D upscaling for unstructured grids. Key technologies are presented and used in a "coarse grid generation" methodology. We describe the iterative construction of geologically complex coarse models that account for sub-scale heterogeneity and anisotropy and that exhibit similar flow response as the reference model. Coarse models capture the fine scale heterogeneity in the form of equivalent permeability tensors and/or upscaled transmissibilities for which computation techniques are proposed and described in this work.

We review next the work of previous researchers in the area of upscaling, reservoir gridding and the use of unstructured grids in reservoir simulation. This literature review highlights the need for a gridding framework that unifies the upscaling, flow adaptation, and assessment of the grid quality and grid robustness procedures. The second chapter presents approaches and results for two-dimensional flow based gridding. Flow-based full-tensor permeability upscaling is extended to unstructured grids and some flow results are presented. In Chapter 3, we extend the streamline method to 3D unstructured grids. The streamline method will then be used to measure efficiently the flow response mismatch due to coarsening. Chapter 4 describes novel techniques for the flow-based upscaling of permeability and transmissibility on 3D unstructured grids, and presents the 3D coarse-grid generation methodology. The tools used to achieve accurate coarse grid generation are

also described. Chapter 5 demonstrates the accuracy and robustness of the new streamline method and the transmissibility upscaling. Chapter 6 illustrates the gridding methodology implemented on synthetic reservoir examples. Finally, in Chapter 7, we draw some conclusions and propose future directions for new investigations and possible improvements.

1.1 Literature Review

This section reviews the relevant literature on three of the topics addressed in this work: the use of unstructured grids in reservoir simulation, the development of absolute-permeability upscaling techniques, and existing approaches for the grid-adaptation of upscaled models. For a literature review on the streamline method and streamline tracing, refer to Chapter 3.

1.1.1 Unstructured grids

Widely used in computational fluid dynamics and other scientific and engineering calculations, unstructured grids are growing in popularity in reservoir simulation (Palagi and Aziz, 1991; Verma and Aziz, 1997; Heinemann *et al.*, 1991, 1998; Gunasekera *et al.*, 1998). A major step in favor of their use was the development of flux-continuous numerical schemes that provided a natural generalization of the well known finite-difference harmonic-mean transmissibility approximation, while also accounting for full tensor permeabilities and grid nonorthogonality (Verma and Aziz, 1997; Edwards and Rogers, 1998; Aavatsmark *et al.*, 1998). These numerical schemes were combined with flow based gridding (Edwards, 2002; Prévost, 2002) to take advantage of the accuracy of flux-continuous full-tensor approximations while saving on the number of grid cells needed to capture the relevant features of the reservoir. The petroleum engineering literature also provides many examples of the use of

quadrilateral and hexahedral grids that rely on the structured nature of the grids while benefiting from some degree of flexibility (Garcia *et al.*, 1992; Edwards *et al.*, 1998; Castellini *et al.*, 2000; Wen *et al.*, 2003a). A possible compromise on the complexity on the grid can be achieved by the use of locally-structured / globally-unstructured grids where the domain is subdivided into hexahedral blocks and structured grids are used in each block (Gunasekera *et al.*, 1998; Jenny *et al.*, 2002). This avenue is promising but may lack generality when considering very complex reservoirs where the definition of the blocks (sub-domains) composing the model may be difficult to achieve. Sub-domain gridding may however prove to be useful when new wells are introduced in the reservoir, therefore limiting the update of the coarse model to one or a few sub-domains (Castellini *et al.*, 2000). A few authors also suggest the use of hybrid (or "heterogeneous") unstructured grids, composed of tetrahedra, hexahedra, prisms and pyramids or, in a 2D context, composed of quadrilaterals and triangles (Edwards, 2002). This approach was successfully applied in 2D but the benefit of the structured grid in terms of solver time may be limited unless special linear solution procedures are applied. Finally, unless it is performed in an automated fashion, the 3D hybrid gridding generally requires editing of the mesh in the regions where transition elements are used (prisms and pyramids) between zones where tetrahedra and hexahedra are used.

The following review focuses on homogeneous unstructured grids, as opposed to locally-structured / globally-structured or hybrid unstructured grids. Heinemann *et al.* (1991) and Palagi and Aziz (1991) were among the first authors to apply Voronoï grids (also called PEBI for perpendicular bisection) to simulate flow in petroleum reservoirs. Their applications for both homogeneous and heterogeneous 2D reservoirs addressed many of the key issues related to grid adaptation, namely: near well refinement, flux discretization, alignment of the grid to permeability trends, and effective permeability calculation (Palagi and

Aziz, 1991). However, those authors relied on scalar permeability values and two-point flux approximations for 2D PEBI grids. Their results demonstrated the benefits of using unstructured grids for the reduction of grid orientation effects and an accurate well geometry resolution.

Verma and Aziz (1997) extended the discretization schemes based on the multiple-point flux approximation developed in Aavatsmark *et al.* (1998) and Edwards and Rogers (1998) to tetrahedral elements. They applied the extended transmissibility calculation to the control volume finite element method on 2D and 3D unstructured grids, allowing full tensor permeabilities and accurate multiple-point flux approximation that no longer depend on a Voronoï or a K-orthogonality grid-criteria. Palagi (1992) exhibited a 2D PEBI grid generation technique based on the use of modules, which was later extended to 3D by Verma (1996). This approach, called modular gridding, constructs the mesh in two steps: (a) the superposition of grid modules, each characterized by user-defined vertex spatial distributions, and (b) the use of a Delaunay tesselation algorithm which connects the vertices, hence defining the property distribution cells as the primal or dual cells. This technique was completed in 2D with the capability of aligning dual cell boundaries with geological features such as faults or internal boundaries. In that case, the geometrical (geological) constraints are enforced in a postprocessing that shifts certain points of the dual grid. However, no indications are given on how the method works at fault intersections, for instance, and a more general method to deal with complex faulting in 3D is still needed.

Gunasekera *et al.* (1998) illustrated the use of the multiple-point flux approximation (MPFA) for a general polyhedral grid. The mesh was composed of hexahedra and tetrahedra. Standard two-point flux was used under the condition of near K-orthogonality. When

MPFA was necessary, additional transmissibility coefficients were provided to the simulator. The paper did not address the issue of grid generation, upscaling or adaptivity to flow information.

Lévy and Prévost (2000) proposed a 2D gridding method based on the joint use of modular gridding (Palagi, 1992; Verma, 1996) and user-defined grid density adaptation. Flow information maps extracted from streamline simulations were used for the adaptation while the gridding technique was based on a metric gridding approach developed by Borouchaki *et al.* (1997). Applications included 2D structured and unstructured grids based on streamline density maps. Some of these results are discussed in Chapter 2.

Hale (2002) advocated the use of unstructured grids from the seismic interpretation to the flow simulation. Combining image processing and optimization techniques, Hale proposed to automatically align a lattice of points with horizons and faults in a 3D seismic image. Connecting these points yields an unstructured grid that can be used both for seismic interpretation and flow calculation. Depending on the type of mesh elements desired, Hale aligned the lattice of "atoms" either on or alongside image features, hence providing cells suitable for either cell- or point- distributed permeabilities. This technique generates the grid in two distinct steps: the optimization of point location (based on point attraction/repulsion information provided by the seismic image) and the mesh generation using a standard Delaunay algorithm. The resulting grid is not constrained in a topological sense by the features on the image, yet the points are introduced so that the cells (primal or dual) conform to the image features. Although this is not illustrated in his paper, the density of the grid can theoretically be adjusted to some information map by spatially varying the value defining the attraction/repulsion potential of the vertices.

Lepage (2002) proposed a framework for gridding complex geological models using

triangular and tetrahedral meshes. The steps of the gridding procedure are as follow. A macro-topology that embeds the various geometrical constraints of the structural model is first devised. A tetrahedral Delaunay refinement algorithm (Shewchuk, 1997, 1998) is then employed to adapt the grid-density to a user-defined resolution constraint. This technique entails solutions to the issues of grid cell quality, grid density and constraints to geological surfaces. Many of the problems that arise in the grid-adaptation to geological features are addressed in this paper. For instance, small angles that arise at pinch-out locations or fault intersections are properly handled (Shewchuk, 2000). Also, grid density information can be readily accounted for both at the surface and the volumetric levels. This technique is different from Borouchaki's metric gridding (Borouchaki *et al.*, 1997) and in particular does not feature its anisotropic adaptation capabilities. Nevertheless, it has proven to be quite robust and is the method of choice for the 3D flow adapted gridding addressed in Chapter 4.

Cao and Kitanidis (1999) presented a methodology for high-accuracy computation of flow in a heterogeneous isotropic formation, employing a dual-flow formulation and adaptive gridding. Their mesh-adaptation approach enhanced the accuracy of the numerical flow solution for a given computational cost. Grid refinements were iteratively introduced using an *a posteriori* error estimator of the finite element solution. This technique specifically addressed reduction in the numerical error. However, it is not guaranteed that the final adapted solution actually converges to the reference solution as the coarse and reference problems (discrete coefficient distributions) are different. This is the case because the cell permeabilities, determined via an upscaling of the underlying geocellular model, change as the grid changes. In the present work, our approach will focus on refinements that improve the definition of the coarse problem relative to the fine description as opposed

to reducing numerical errors, which are believed to be second order effects in many of the cases that are of interest to us.

Most of the gridding techniques described above are practically applicable to obtain reasonably accurate reservoir descriptions in 2D or (to a lesser extent) in 3D. However, only the framework devised by Lepage (2002) (based on the work of Shewchuk, 1997) enables a systematic accounting of both geological features and spatially varying grid quality criteria. A significant flexibility of the grid adaptation is left to the user and, as will be demonstrated, several quality criteria can be devised and integrated in the gridding procedure. As a result of this, the grid generation algorithms are tailored to produce quality flow models.

1.1.2 Upscaling

The most fundamental requirement of the coarse model is that it should reproduce in an average sense the flow response of the fine model. Hence, a critical aspect of the coarsening process is the accurate calculation of equivalent permeabilities or transmissibilities such that the discretized coarse scale flow equations capture the sub-cell heterogeneity. This subsection summarizes approaches used in the literature to compute equivalent permeability representations from detailed permeability descriptions. For comprehensive reviews, refer to Renard and de Marsily (1997) and Wen and Gómez-Hernández (1996). See He *et al.* (2002) for the description of a permeability upscaling technique on general quadrilaterals using a mixed finite element solution. See Eek-Jensen *et al.* (1999) for applications of a 3D upscaling technique on general quadrilaterals and Trykozko *et al.* (2001) and Zijl and Trykozko (2001) for recent developments in upscaling using nodal and mixed finite element methods.

General discussion of upscaling techniques

We now consider the determination of equivalent permeability coefficients for the pressure equation. The equivalence of flow at the two different scales is generally based on the assumption of steady-state, uniform[1], single-phase flow. In the following, the coarse scale permeability is referred to as the *effective* or *equivalent* permeability while the permeability from the detailed description is called the *fine* scale permeability.

Effective permeabilities can be viewed as "averaged" values. However, since permeability is not an additive property, this analogy can be misleading. Several analytical results for the effective permeability are available. For flow in a layered reservoir, the effective permeability for flow across the layers is exactly the weighted harmonic mean of the layer permeabilities. The arithmetic average is the result for flow along the layers. For an isotropic checker board pattern or for an isotropic lognormal permeability distribution, the effective permeability is equal to the geometric average. Also, Cardwell and Parson (1945) proved that arithmetic and harmonic averages are respectively the upper and lower bounds of the effective permeability. Based on this observation, Journel *et al.* (1986) and Deutsch (1989) proposed a continuum of power-averaging methods ranging between the harmonic and arithmetic extremes. Although very efficient computationally, power-averaging methods have some limitations. Specifically, the value of the power exponent is essentially case-dependent and needs calibration. In addition, these methods cannot be expected to be as accurate as numerically based procedures.

Because permeabilities at the fine scale are generally assigned to a volume-support larger than the volumes of the measurement scale, the fine permeability is possibly already a tensor. For spatially correlated media, effective permeability is generally a tensor even

[1] as opposed to radial for instance

when the fine permeability is isotropic. The power-averaging methods do not solve for the tensor principal directions but, if those are provided, power averaging can be applied to calculate the components of the tensor.

Discussion of flow-based upscaling

Many upscaling methods (*e.g.*, so-called *analytical* methods) make limiting assumptions on either the non-tensorial aspect of the upscaled permeability (power averages, renormalization) or on the periodic character of the fine permeability distribution (homogenization theory). However, numerical experiments showed that from a flow perspective, the equivalent permeability or transmissibility can depend on the boundary conditions applied on the domain. This dependence is generally strongest for blocks of size comparable or smaller than the correlation length of the fine permeability. This observation has led to a large literature of *numerical* upscaling methods. These methods are based on the resolution of the pressure equation on a domain including the block of interest for certain sets of boundary conditions and are hence referred to as *flow-based* methods. Flow-based methods essentially differ from one another in the type of boundary conditions applied, the size of the region simulated, and the way in which the upscaled property (*e.g.*, permeability) is computed from the fine scale simulation. Depending on the region considered in the flow simulation compared to the target region for which the permeability is sought, flow-based methods can be classified as either local, extended-local or global.

Warren and Price (1961) were the first to apply flow-based methods. They solved the single phase flow problem to provide effective permeabilities in each direction of space. White and Horne (1987) solved for the upscaled transmissibility by solving multiple global fine flow problems and applying a linear least square regression to fine scale pressures

and fluxes. They appear to be the first researchers to consider the full-tensor aspect of the upscaled permeability.

Durlofsky (1991) proposed a method that solves for a general permeability tensor and also guarantees that it is positive-definite. This property comes from the use of periodic boundary conditions in the resolution of the pressure equation. Wen *et al.* (2000, 2003b) showed that flow-based upscaling using either border-regions or a rotation of the domain under study both produced improved results in some cases over the purely local upscaling presented by Durlofsky (1991). Their idea makes use of the 'skin' introduced by previous investigators but in a periodic boundary context. A possible limitation of the technique is that the positive-definiteness of the equivalent permeability tensor is no longer guaranteed. This is recovered approximately through use of a least square procedure (Wen *et al.*, 2003b).

He *et al.* (2002) proposed an upscaling technique for general quadrilateral cells. Their method entails the mixed finite element solution of the local fine scale pressure equation over the quadrilateral region. The heterogeneities internal to the quadrilateral are resolved using triangular elements and periodic boundary conditions are applied. The resulting permeability tensor is computed via integration of the fine scale velocity and pressure gradients. The principal difficulty of the method lies in the application of periodic boundary conditions on non-rectangular cells. For rectangles, quadrilaterals, or parallelograms, the proposed method is equivalent to the one of Durlofsky (1991) which was shown to be improved by the use of border regions (Wen *et al.*, 2000, 2003b). The use of border regions or alternate boundary conditions within the context of the method of He *et al.* (2002) is also possible.

Edwards (2002) presented an application of the border-region upscaling for 2D unstructured grids by averaging velocities and pressure gradients over a general polygon instead of a rectangle. The resulting tensors were used in a flux-continuous numerical scheme for unstructured grids.

Extension of the flow-based upscaling techniques to 3D unstructured grids does not appear to have been reported in the literature. Extension of the approaches presented for the 2D case is not trivial and may require sophisticated procedures to achieve an efficient and accurate upscaling.

Many of the procedures described above can also be used to compute upscaled transmissibilities. In some cases, the use of upscaled transmissibility has been shown to provide coarse scale simulation results of better accuracy than those obtained with upscaled permeability (*e.g.*, Chen *et al.*, 2003). To date, most transmissibility upscaling procedures have been applied within a two-point flux context, though upscaled transmissibilities can be used within a multi-point flux setting. In this work, we will apply both permeability and transmissibility upscaling for 3D unstructured grids.

1.1.3 Grid adaptation

By grid adaptation, we mean the adaptation of the simulation grid based on information related to static properties of the model (like permeability or internal boundaries) or dynamical flow-related information. In the latter case, we use the term ‘flow-adaptation’. By contrast, dynamic grids and dynamic local grid refinements that track saturation fronts and evolve during the simulation are *not* discussed in this work, as the ‘static’ grid approach is still the norm in reservoir simulation.

Flow adaptation

Several authors recognized that the quality of the reservoir model is affected by the orientation of the reservoir features relative to the simulation grid. Features that are not aligned with the x, y, z axes of a Cartesian grid can be modeled in different ways, either by accurately computing equivalent permeability tensors by locally rotating the grid for instance (Wen *et al.*, 2000), or by locally distorting the grid to 'track' the heterogeneity trends (Verma and Aziz, 1997; Edwards *et al.*, 1998; Castellini *et al.*, 2000; Wen *et al.*, 2003a).

Verma and Aziz (1997) used streamlines as a curvilinear coordinate to construct a structured quadrilateral grid. Edwards *et al.* (1998) further developed the technique of streamline gridding by also including isopotential lines orthogonal to the streamlines. The principal improvement offered by this gridding technique is the reduction in pressure discretization complexity resulting from diagonal permeability tensors in the local coordinate system.

Castellini *et al.* (2000) extended the technique introduced in Verma and Aziz (1997) and Edwards *et al.* (1998) by performing the flow adapted gridding in a modular context. The domain was decomposed into sub-domains (modules) such that the flow-based grid generation could be performed locally. Unfortunately, the sub-domain definition is not based on flow information and may artificially set non-K-orthogonal boundaries. As a consequence, this technique may provide flow-results which are inferior to those obtained using a global technique.

Lévy and Prévost (2000) combined modular gridding approaches (Verma and Aziz, 1997; Palagi and Aziz, 1991) and user-defined density maps for structured and unstructured flexible grids in 2D. Prévost *et al.* (2001) gave an example of the use of streamlines for the generation of 2D adapted unstructured grids as an extension of previous structured

streamline grid generation (Verma and Aziz, 1997).

Coarsening procedure

Durlofsky *et al.* (1996, 1997) described a technique to use flow-rates obtained on the global (or sub-global) domain to achieve the scale-up of layered reservoirs using nonuniform coarsening. This technique exhibited reasonable robustness to well placements and conditions while considerably reducing the computing time. The coarsening can be performed in each direction of space in sequence and relies strongly on the structured nature of the grid and some degree of layering in the permeability field.

Li and Beckner (2000) proposed an uplayering method based on the iterative placement of coarse layers. The criteria of optimality is based on a minimization of the variance of a mean property computed layer-wise. They proposed using layer properties such as mean permeability and static measures of the dispersion of multiphase flow. This method can be extended in a straightforward manner to also account for mean flow rates within layers for instance. However, the method as implemented did not use flow information.

Garcia *et al.* (1992) proposed an "elastic" grid adjustment method to generate structured non-orthogonal grids. Developed both for 2D and 3D heterogeneous reservoir models, this technique iteratively minimizes a measure of sub-grid heterogeneity in order to reduce the impact of permeability upscaling, control the shape and sizes of the blocks, and achieve grid refinement in specific regions of the reservoir. Alternate criteria can be used for the minimization and the use of flow rates was suggested by Tran (1995) and Wen (1996). Elastic gridding techniques provide the benefit of retaining the computationally-efficient structured nature of the grid. Even though the restrictions of i, j, k indexing of the grid is somewhat relaxed by the use of corner point geometry (CPG) cells, the scheme still

lacks full flexibility. Indeed, a greater cell-density in a given region can either be achieved by "shifting" nodes from neighboring regions (constant grid size) or by inserting a new plane of cells, therefore globally changing the grid. For instance, in 3D, an insertion of a plane of cells in the i direction of a $n_i \times n_j \times n_k$ grid results in $n_j \times n_k$ additional cells. These cells span over a region that is potentially much larger than the one targeted by the refinement, thus impacting the effectiveness of the insertion. Other concerns may arise from the distortion of the CPG cells, as the cell-vertices are allowed to move independent of geometrical constraints.

Some guidelines for the generation of structured 3D coarse models exist in the literature. When coupled with an accurate upscaling procedure, these methodologies can, in many cases, provide reasonable coarse models for various well configurations. Flow-adapted unstructured-grid generation has been attempted (Prévost *et al.*, 2001; Cao and Kitanidis, 1999) for 2D models, though a general methodology for 3D unstructured grid generation accounting for complex geometries and using accurate upscaling and flow response quality control is still needed.

1.2 Statement of Purpose

It is recognized from the above literature review that there is a strong need for a clear procedure to generate coarse models that capture flow-relevant features and that is applicable to complex reservoir geometries in 3D. Examples of such procedures were presented above in the context of structured grids but the power of grid adaptation will only be complete when used in conjunction with unstructured 3D grids. An important requirement for coarse reservoir modeling is the prior description of a flow diagnostic for the reservoir. This diagnostic must be efficiently obtained for both the fine scale detailed reservoir and

the generated coarse models. This flow-diagnostic provides the limits of acceptable coarse models and should capture the impact of the fine features on the flow-responses. Of course, the notion of relevance of the reservoir features is relative to the type of displacements that are to be modeled for the reservoir under study. As a consequence, some assessment of the robustness of the proposed methodology is also needed.

This work proposes a methodology for coarse, unstructured-grid reservoir-model generation that includes:

1. the gridding procedures,

2. the calculation of equivalent coarse scale properties,

3. the efficient assessment of flow response for unstructured grids,

4. the ability to iterate through the grid-construction process.

Regarding the first point, gridding procedures have made significant progress in recent years and advances in reservoir gridding are embodied in research tools such as the TGrid-Lab plug-in from the Gocad research consortium[2]. This work proposes to make direct use of these new techniques, although the linkage of these methods with flow information has not been established. This work will address this important issue. Points (2) and (3) concern technologies which are not available as of today. The present work proposes novel and efficient solutions to these problems. Finally, the overall methodology for building a coarse model and iterating on its construction until acceptable results are obtained is presented as a generalization of previous approaches for structured grids.

[2]The software platform used in this work is the "Gocad developer kit Release 2.0.7", made available by Earth Decision Sciences (http://www.earthdecisionsciences.com). Research plugins provided by the "Laboratoire Infographie et Analyse des Données" (http://gocad.ensg.inpl-nancy.fr)

Chapter 2

2D Adapted gridding

2.1 Introduction

When handling complex geometry, reservoir simulation using structured grids can require a large number of cells due to the difficulty of conforming to boundaries, faults and pinch-outs as well as the constraints on grid-cell size variations within the model. Furthermore, these limitations restrict the magnitude of upscaling that is possible. The natural adaptivity of unstructured grids in terms of shape and element sizes is a great asset that is currently underexploited in reservoir simulation. While several authors (Heinemann *et al.*, 1991; Palagi and Aziz, 1991; Verma and Aziz, 1997; Heinemann *et al.*, 1998) have demonstrated the use of unstructured grids to adapt to the geometry, we currently lack clear criteria for adaptivity of the elements to boundary effects and underlying property distributions. The accurate representation of the effects of sub-scale heterogeneity at the coarse grid level is critical to the accurate approximation of the response of the fine scale model. Hence, accurate computation of the effective (or equivalent) permeability as well as appropriate grid-densification in important regions of the domain are needed.

The first part of this chapter addresses the issues related to gridding and grid-adaptivity. Next, the generalization of the equivalent permeability tensor calculation for unstructured grids, as well as the discretization scheme used in this work, are presented. This is followed by the description of a fast flow-response evaluation, based on streamline simulation. Finally, results are presented for several cases which demonstrate the applicability and accuracy of the unstructured gridding and upscaling methodology in 2D.

2.2 Unstructured grids

Subdivision of a domain into cells is needed when discretizing the reservoir flow equations. This can be achieved via structured or unstructured grids. In the first case, the topology that describes the neighborhood relationships between the simulation cells is implicit. The classic i, j, k indexing scheme used in structured grids (Figure 2.1) provides banded connectivity matrices resulting in efficient numerical methods. Furthermore, the geometry of the cells can depart significantly from the strict Cartesian (or orthogonal) case to gain some geometrical flexibility while retaining the indexing structure, as shown in Figure 2.2. However, many limitations to the use of structured grids still exist, among which are the difficulty to conform to exterior or interior boundaries and the inherent lack of control of cell sizes (Figure 2.3). For complex geometry domains or when a greater adaptivity of the cell size is needed, a natural solution is to use unstructured grids. In that case, the adjacency relation between cells is completely general, resulting in greater domain boundary conformity and cell size flexibility.

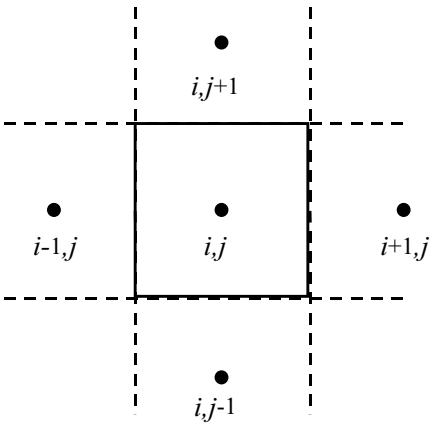


Figure 2.1: Implicit ordering scheme in a structured grid

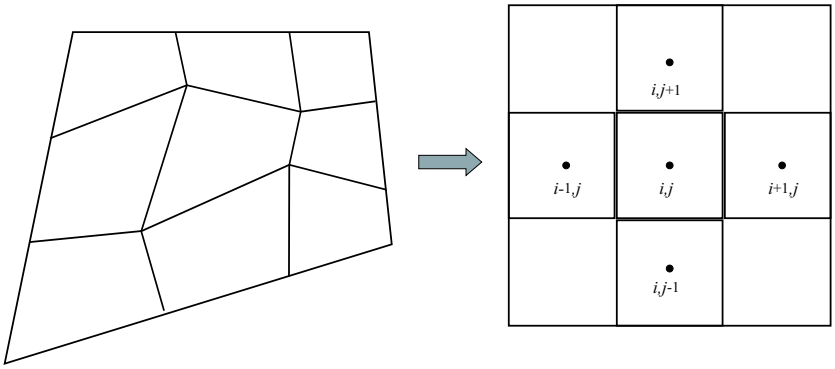


Figure 2.2: Logically rectangular grid which is non-Cartesian

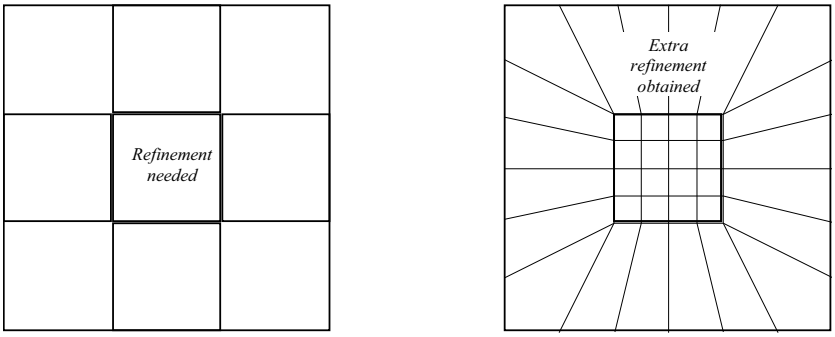


Figure 2.3: The structured nature of Cartesian and corner point geometry grids can result in unnecessary grid refinement

2.2.1 Simplex, Duality and Tessellation

We use in this work some vocabulary borrowed from disciplines such as topology and mesh generation. A simplicial grid designates a subdivision of space in the most 'simple' elements (simplex). In an n-dimension space, these elements have $n + 1$ vertices. A tesselation refers to the construction of a simplicial grid from a given set of vertices. This operation is always possible in 2D (Delaunay triangulation algorithm) but may require the extension of the original set of vertices in 3D. These additional points are called Steiner points.

Generalization of the cell-centered structured grids to unstructured grids results in polygonal (2D) and polyhedral (3D) cells. Polygonal and polyhedral grids can be viewed as the dual grid obtained from a primal simplicial grid (Figure 2.4). The simplex, composed of triangles in 2D and tetrahedra in 3D, can be obtained by a process called Delaunay tessellation. Simplices obtained via Delaunay tessellation have several advantages including the "empty sphere" and the "largest minimum-angle" properties that optimize the shape of the elements of the grid among all possible tessellations for a given set of vertices. From a flow simulation perspective, the dual grid plays an important role in the control-volume finite-element method formulation. In some approximations, it defines the volume support for the rock and fluid properties. For further explanation, refer to Section 3.2.2.

Tessellations can be constrained to honor exterior and interior boundaries to guarantee that no edge (face in 3D) intersects such boundaries. Two approaches are commonly considered when dealing with interior geometrical constraints. The first approach, called *constrained Delaunay tesselation*, consists in honoring exactly the nodes of the constraints without any further node insertion. In the case of 2D meshes, the constraints are lines and a

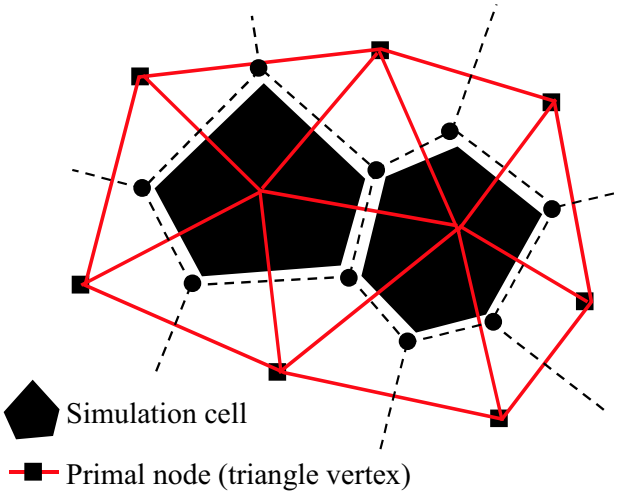


Figure 2.4: Triangulation and corresponding Voronoï dual grid

tesselation can always be obtained from the constraints and a given set of vertices. This process does not require any point insertion but generally leads to non-Delaunay tesselations and hence to the loss of the Delaunay properties around internal and external boundaries. A second possible approach is called the *conforming Delaunay tesselation*. In this approach, the vertices defining the constraints can be augmented to honor the constraint geometry while also enforcing the Delaunay criteria (see Shewchuk, 1997; Lepage, 2002).

In 3D, the constraints involve triangulated surfaces and constrained tesselation consists in honoring the geometry of the surfaces as well as the surface topology (a triangle on a constraining surface must be a face of the tetrahedral mesh). However, constructing such a tesselation is generally not possible without the insertion of Steiner points (see Section 4.3 for a discussion of 3D constrained gridding).

2.2.2 Tessellation constrained by a metric

Constraints on the shape and area of the elements can also be enforced away from boundaries by performing the tessellation according to a metric (Borouchaki *et al.*, 1997). A scalar property map can, for instance, be used to constrain the element size (in that case we will call it a target resolution). The points inserted in the grid are then optimally selected to match the density constraint. We will exploit this capability later in this chapter, when we use flow maps to guide grid density. In addition to the control on the grid density, the metric constraint can be used to control the anisotropy of the grid. This may result in elements that are skewed but which may provide better condition numbers of the resulting matrix and which may in turn lead to an overall reduction in computation. Furthermore, the skewed shape of the elements may result in a more efficient distribution of the cells, hence reducing the problem size.

2.3 Full-tensor permeabilities

Accurate representation of the effect of the fine scale permeability on the coarse model is important to reproduce essential flow-response characteristics. This requirement implies the need for the accurate calculation of an equivalent permeability tensor for each coarse grid cell. Accounting for the tensorial aspect of permeability entails the use of upscaling techniques that compute the off-diagonal elements of the permeability tensor and the use of finite volume stencils that are able to accommodate full-tensor permeabilities.

2.3.1 Calculation of equivalent permeability tensors

Current methods to compute effective permeability values include power averaging, renormalization, and the solution of local flow problems (see Chapter 1 as well as Wen and Gómez-Hernández (1996) and Renard and de Marsily (1997) for reviews). In the case of flow-based methods, for any given coarse grid cell a sub-domain of the fine model including the target coarse cell is considered. A series of single-phase incompressible flow problems are solved on this sub-model with appropriate boundary conditions. Finally, the coarse level equivalent permeability $\mathbf{k}^*$ is computed by requiring the same mean flow rate for the same mean local pressure drop on the fine and coarse scales.

The two equations that describe steady-state, single-phase flow in porous media are Darcy's law and mass balance equations:

$$\begin{aligned} \mathbf{u} &= -\Lambda \mathbf{k} \nabla P \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \tag{2.1}$$

where $\mathbf{k}$ $[\mathrm{m}^2]$ is the permeability tensor, Λ $[1/\mathrm{Pa.s}]$ is the fluid mobility ($\Lambda = 1/\mu$ where μ is the viscosity), $\mathbf{u}$ $[\mathrm{m/s}]$ is the fluid velocity and P [Pa] is the pressure. Combining the two equations of the system 2.1 leads to the single phase pressure equation:

$$\nabla \cdot \Lambda \mathbf{k} \nabla P = 0 \tag{2.2}$$

Defining the equivalence between a fine heterogeneous medium and homogeneous medium at the coarse scale is only possible in a limited sense. A possible criteria of equivalence is the equality of flow at the boundaries between the heterogeneous medium and

the equivalent homogeneous media under the same pressure difference (Cardwell and Parson, 1945). We will prefer a volume average equivalence criteria proposed by Rubin and Gómez-Hernandéz (1990):

$$\frac{1}{V}\int_V \mathbf{u}(\mathbf{x})\ dV(\mathbf{x}) = -\Lambda\mathbf{k}^*\left(\frac{1}{V}\int_V \nabla P(\mathbf{x})\ dV(\mathbf{x})\right) \tag{2.3}$$

The use of periodic boundary conditions to solve the local problem (Durlofsky, 1991), in conjunction with the use of a border region (Wen *et al.*, 2003b) around the target cell, was shown to capture the effects of anisotropy and connectivity with reasonable accuracy in a number of cases. Other boundary conditions, or a local-global procedure (Chen *et al.*, 2003) can also be applied and may be preferable in some cases.

If we consider a periodic domain where the augmented cell is replicated in each coordinate direction and further assume that the flow is periodic (which is often a reasonable assumption away from sources / sinks), the flow in the interior of the original cell can be obtained by solving a local flow problem with appropriate periodic boundary conditions. Periodic boundary conditions impose the pressure field on the domain under study to be of the form:

$$P = P_0 + \mathbf{G}\cdot(\mathbf{x}-\mathbf{x}_0) \tag{2.4}$$

where $\mathbf{G} = G_x\mathbf{i}_x + G_y\mathbf{i}_y$ is an arbitrary constant vector, P_0 an arbitrary pressure, and $\mathbf{x}_0$ is a reference location.

In the case of a Cartesian coarse grid, following the notation of Wen *et al.* (2003b), the border region is constructed by augmenting the initial (or target) system by a certain number of rings. Each ring is composed of the fine cells contained in a layer of coarse cells added around the central coarse block. An example with one such ring ($r = 1$) is shown in

Figure 2.6.

The boundary conditions are then applied on this augmented system:

$$\begin{aligned} P(x,0) &= P(x,l_y) - G_y l_y && \text{on } y = 0 \text{ and } y = l_y \\ P(0,y) &= P(l_x,y) - G_x l_x && \text{on } x = 0 \text{ and } x = l_x \\ u_y(x,0) &= u_y(x,l_y) && \text{on } y = 0 \text{ and } y = l_y \\ u_x(0,y) &= u_x(l_x,y) && \text{on } x = 0 \text{ and } x = l_x \end{aligned} \tag{2.5}$$

where l_x and l_y designate the length of the augmented system (see also Figure 2.5).

In the case of an unstructured grid, the same technique is used and a border region around the control volume can be generally defined as a rectangular subset of fine cells that includes the target control volume. The notion of a border region can therefore be defined for unstructured grids as a rectangular set of cells containing the triangles connected to the center of the control volume (see Figure 2.7). The same boundary conditions as above can be applied.

For both the structured and the unstructured cases, Equation 2.2 subject to the boundary condition 2.5 must be solved twice over the extended region. In the first solution, we specify $G_x^1 = 1$ and $G_y^1 = 0$ and in the second solution $G_x^2 = 0$ and $G_y^2 = 1$. Then, as required in Equation 2.3, we compute the average velocity and pressure gradients for each flow problem j:

$$\begin{aligned} \langle \mathbf{u} \rangle^j &= \frac{1}{V} \int_V \mathbf{u}^j \, dV \\ \langle \nabla P \rangle^j &= \frac{1}{V} \int_V (\nabla P)^j \, dV \end{aligned} \tag{2.6}$$

where V refers to the volume of the target coarse cell; *i.e.*, the center Cartesian cell in the structured case or the control volume in the unstructured case.

The equivalent permeability tensor $\mathbf{k}^*$ is therefore constrained by the following linear

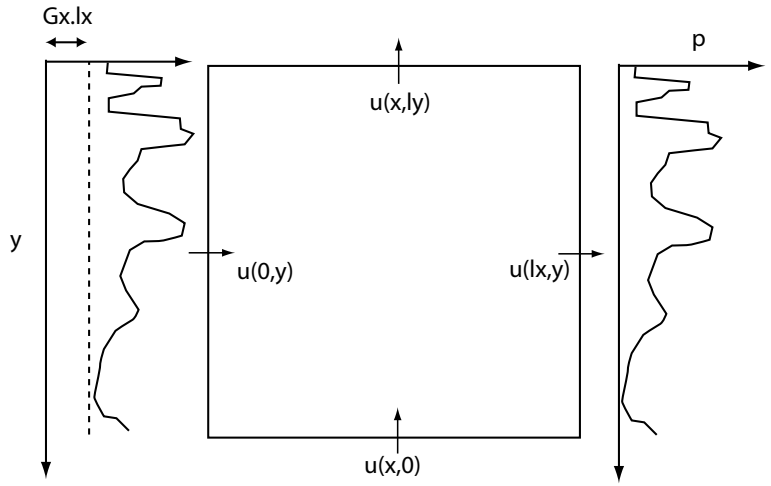


Figure 2.5: Periodic boundary conditions

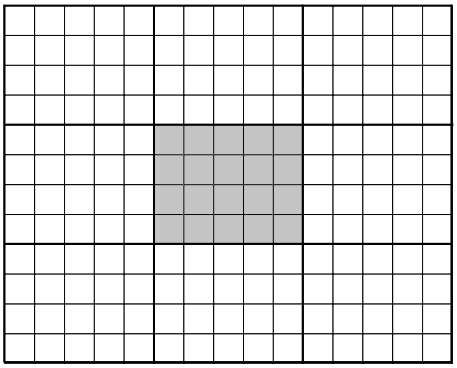


Figure 2.6: Target coarse Cartesian cell (in gray) augmented by an $r = 1$ border region

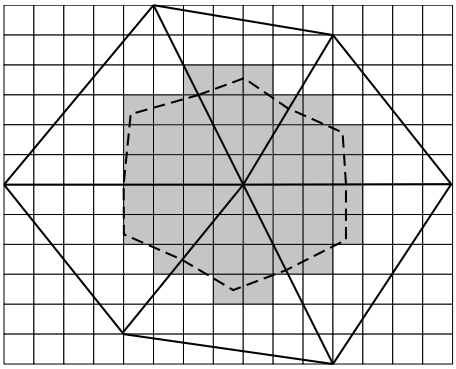


Figure 2.7: Target coarse cell (in gray) augmented by a border region

system involving the simulated average flow properties and the unknown tensor components:

$$\begin{bmatrix} \langle \partial P/\partial x \rangle^1 & \langle \partial P/\partial y \rangle^1 & 0 & 0 \\ 0 & 0 & \langle \partial P/\partial x \rangle^1 & \langle \partial P/\partial y \rangle^1 \\ \langle \partial P/\partial x \rangle^2 & \langle \partial P/\partial y \rangle^2 & 0 & 0 \\ 0 & 0 & \langle \partial P/\partial x \rangle^2 & \langle \partial P/\partial y \rangle^2 \end{bmatrix} \begin{bmatrix} k^*_{xx} \\ k^*_{xy} \\ k^*_{yx} \\ k^*_{yy} \end{bmatrix} = -1/\Lambda \begin{bmatrix} \langle u_x \rangle^1 \\ \langle u_y \rangle^1 \\ \langle u_x \rangle^2 \\ \langle u_y \rangle^2 \end{bmatrix} \tag{2.7}$$

In the general case, the resulting tensor is full (non-zero off-diagonal terms), even when the fine scale permeability is a scalar or a diagonal tensor. A possible drawback of the use of a border region is the loss of guaranteed symmetry and positive definiteness. In addition to the linear constraints, the full tensor defined via Equation 2.7 may also be constrained to be positive definite and symmetric. Symmetry can be recovered by either averaging the off-diagonal terms resulting from Equation 2.7, or by introducing an extra linear equation forcing $k_{xy} = k_{yx}$ and solving the overdetermined system in a least square sense (Wen *et al.*, 2003b). In this work, asymmetry is corrected by averaging. Positive definiteness is in practice rarely an issue. If not satisfied, positive definiteness is trivially enforced by re-performing the flow based upscaling without border regions on the smallest rectangular domain enclosing the target control volume.

We note that valid permeability tensors can also be obtained using the same border region method but by applying other boundary conditions. Which type of boundary conditions, and how large should the border region be, remain somewhat open (and presumably case dependent) questions. In many cases, however, the differences in results obtained from the various boundary conditions are relatively slight.

2.3.2 Full k discretization

Numerical schemes for the discretization of the pressure equation using finite difference techniques with full tensor permeabilities were presented on Cartesian and corner point geometry grids by Edwards and Rogers (1998), Aavatsmark *et al.* (1996, 1998) and Lee *et al.* (1998, 2002). Extensions to control volume finite elements generalize the notion of flux-continuous schemes to triangular and tetrahedral meshes (Aavatsmark *et al.*, 1996; Verma and Aziz, 1997; Edwards and Rogers, 1998). The technique entails the solution of local linear problems on the coarse grid enforcing pressure and flux continuity at the cell interfaces. The unknowns of the linear systems are the cell-to-cell transmissibilities, which are then used in the global solution. Below is a summary of the derivations leading to the flux continuous CVFE scheme for 2D grids. Complete derivations for the 3D case can be found in Verma and Aziz (1997).

Let us consider a triangle $\mathcal{F}$ (Figure 2.8). $\mathcal{F}$ can be divided into three regions corresponding to the three control volumes sharing its area. For each region i, the pressure $P^i(x, y)$ is assumed to vary linearly:

$$\nabla P^i = \begin{bmatrix} \partial P^i/\partial x \\ \partial P^i/\partial y \end{bmatrix} \tag{2.8}$$

$$\nabla P^0 = \begin{bmatrix} (x_b - x_0) & (y_b - y_0) \\ (x_c - x_0) & (y_c - y_0) \end{bmatrix}^{-1} \begin{bmatrix} P_b - P_0 \\ P_c - P_0 \end{bmatrix} \tag{2.9a}$$

$$\nabla P^1 = \begin{bmatrix} (x_a - x_1) & (y_a - y_1) \\ (x_c - x_1) & (y_c - y_1) \end{bmatrix}^{-1} \begin{bmatrix} P_a - P_1 \\ P_c - P_1 \end{bmatrix} \tag{2.9b}$$

$$\nabla P^2 = \begin{bmatrix} (x_a - x_2) & (y_a - y_2) \\ (x_b - x_2) & (y_b - y_2) \end{bmatrix}^{-1} \begin{bmatrix} P_a - P_2 \\ P_b - P_2 \end{bmatrix} \tag{2.9c}$$

Using Darcy's law, the piecewise constant expression for the velocity given a permeability tensor $\mathbf{k}$ is

$$\mathbf{u}_i = -\Lambda \mathbf{k}_i \begin{bmatrix} \partial P^i / \partial x \\ \partial P^i / \partial y \end{bmatrix} \qquad i \in \{0, 1, 2\} \tag{2.10}$$

While pressure continuity is embedded in the system (one pressure defined at a, b, and c), flux continuity must be enforced. This is accomplished by writing:

$$\begin{cases} \mathbf{u}_0 \cdot \hat{\mathbf{n}}_\mathbf{b} = \mathbf{u}_2 \cdot \hat{\mathbf{n}}_\mathbf{b} \\ \mathbf{u}_0 \cdot \hat{\mathbf{n}}_\mathbf{c} = \mathbf{u}_1 \cdot \hat{\mathbf{n}}_\mathbf{c} \\ \mathbf{u}_1 \cdot \hat{\mathbf{n}}_\mathbf{a} = \mathbf{u}_2 \cdot \hat{\mathbf{n}}_\mathbf{a} \end{cases} \tag{2.11}$$

Combining Equations 2.8, 2.10 and 2.11, a 3×3 linear system can be written expressing the intermediate pressures as a function of P_0, P_1 and P_2.

For instance (using the notation shown in Figure 2.8), the flux continuity condition

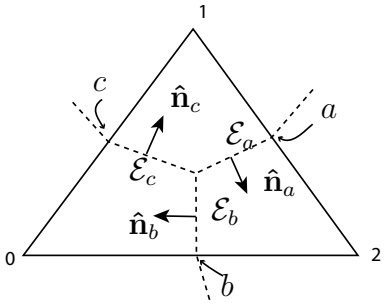


Figure 2.8: The transmissibilities are obtained by introducing a piecewise linear pressure in each region i and enforcing both pressure continuity at the points a, b and c and flux continuity across the edges $\mathcal{E}_a$, $\mathcal{E}_b$ and $\mathcal{E}_c$

across the edge $\mathcal{E}_b$ reads:

$$\left(\mathbf{k}_0 \begin{bmatrix} (x_b - x_0) & (y_b - y_0) \\ (x_c - x_0) & (y_c - y_0) \end{bmatrix}^{-1} \begin{bmatrix} P_b - P_0 \\ P_c - P_0 \end{bmatrix} \right.$$
$$\left. -\mathbf{k}_2 \begin{bmatrix} (x_a - x_2) & (y_a - y_2) \\ (x_b - x_2) & (y_b - y_2) \end{bmatrix}^{-1} \begin{bmatrix} P_a - P_2 \\ P_b - P_2 \end{bmatrix} \right)^T \cdot \hat{\mathbf{n}}_b = 0$$

with $\mathbf{k}_i$, the permeability tensor at the node (control volume) $i \in \{0, 1, 2\}$.

Similar constraints for edges $\mathcal{E}_a$ and $\mathcal{E}_c$ can be written and the system of flux continuity is obtained:

$$A \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} = B \begin{bmatrix} P_a \\ P_b \\ P_c \end{bmatrix}$$

which must then be symbolically inverted. This leads to the following matrix form:

$$\begin{bmatrix} P_a \\ P_b \\ P_c \end{bmatrix} = \begin{bmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix} \tag{2.12}$$

Then, using Equations 2.9, the Darcy velocities in each region can be expressed as linear combinations of the pressure at the triangle nodes (centers of the control volumes). For instance, for region $i = 0$:

$$\mathbf{u}_0 = -\Lambda \mathbf{k} \begin{bmatrix} (x_b - x_0) & (y_b - y_0) \\ (x_c - x_0) & (y_c - y_0) \end{bmatrix}^{-1} \begin{bmatrix} R_{10} - 1 & R_{11} & R_{12} \\ R_{20} - 1 & R_{21} & R_{22} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \end{bmatrix}$$

Finally, the fluxes q_a, q_b and q_c can be expressed in terms of the pressures, giving the algebraic expression of the transmissibility coefficients:

$$q_a = \sum_{i=0}^{2} T_{ai} P_i \tag{2.13}$$

where T_{ai}, $i \in \{0, 1, 2\}$ are the transmissibility coefficients describing the flux across the edge $\mathcal{E}_a$.

2.4 Fast reservoir evaluation using streamline method

When considering a large detailed reservoir model, the coarse grid model should be able to reproduce the essential fine model flow characteristics. For this purpose, it is critical to be able to evaluate and compare flow responses of the coarse and fine reservoir models.

Although a fine-scale multiphase flow simulation is not practical (that is precisely why the upscaling step is needed), the resolution of a single (steady state) pressure solution may still be affordable, using a parallel solver or domain decomposition if necessary. In that case, it is possible to extract flow data such as total flow rates for given boundary conditions, as well as displacement information such as production history and saturation maps. Using streamline simulation, this information can be effectively and accurately obtained for both structured and unstructured grids (Prévost *et al.*, 2002).

Streamline simulation (Batycky *et al.*, 1997) decouples the pressure equation from transport by summing the conservation equations, removing $\Delta_t S$ terms from the pressure equation, and treating the saturation dependency of transmissibility explicitly. Moreover, the transport equations are written in terms of one-dimensional problems along the streamlines. As a consequence, this formulation is somewhat analogous to the IMPES method (see Breitenbach *et al.* (1969) and Coats (1979) for the derivation of the three-phase IMPES equations).

A very advantageous speedup is obtained by assuming that the streamlines vary slowly in time (small effect of the change in saturation on the pressure distribution). In the case of an incompressible tracer flow simulation (unit mobility ratio displacement), this assumption is exact. Thus, the production history of every well can be obtained using only one pressure solve (assuming no change in operating conditions) as shown on Figures 2.9, 2.10 and 2.11. The producing tracer fractional flow at a given time is equal to the fraction of streamlines that have broken through to the producer at that time.

From the above discussion, it is clear that streamline simulation provides a fast and accurate means for assessing the accuracy of the coarse model relative to the reference fine scale model. While the assumption of tracer flow does not incorporate the non-unit

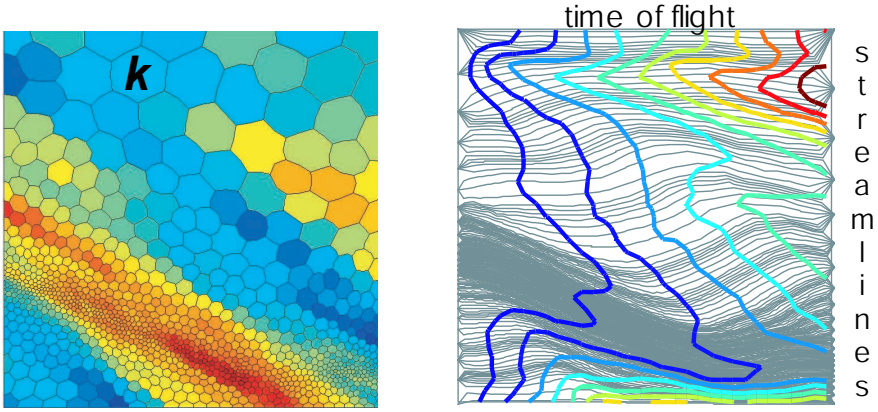


Figure 2.9: Permeability field and left to right flow simulation (streamlines and isovalues of the time of flight)

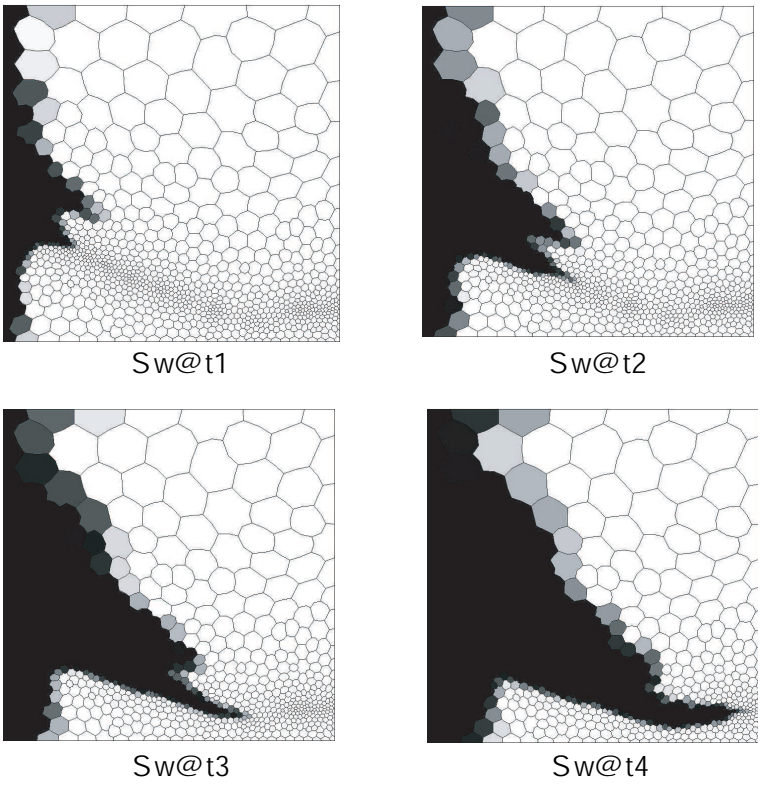


Figure 2.10: Water cut and saturation fields for the problem described in Figure 2.9

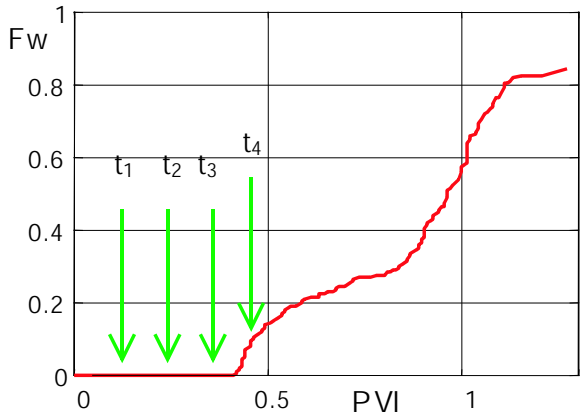


Figure 2.11: The time of flight contours give the position of the tracer front at a given simulation time

mobility ratio effects in the coarse to fine model comparison, it allows us to extract a first approximation of the impact of the upscaling on the accuracy of the displacement calculation. This comparison is provided at the cost of a single pressure solution. The extension of the streamline method to 2D and 3D unstructured grids is described in Chapter 3.

2.5 Application to two-dimensional grid generation

Unstructured grids, streamline simulation and flow-based upscaling can be used in combination to generate coarse reservoir models. While the detailed description provides a reference solution for a unit-mobility displacement, information maps obtained from fine-scale flow information serve as an adaptation criterion for the coarse grid generation. The coarse grid is hence chosen such that it “minimizes” the error introduced by the coarsening for different boundary conditions. The current section presents two examples of coarse grid adaptation to detailed flow results. In all the cases presented, streamline simulation is used to assess the improvement of the adapted coarse grid over a uniform coarse grid. The first example demonstrates the use of a simple refinement criterion to adapt the grid to the facies

distribution in the reservoir. The second example shows the use of a target grid resolution to constrain the grid density to particular information (or density) maps. Here, two alternative maps are considered to minimize local discrepancies in the production fractional flow curves (water cut): a mean velocity map and a streamline breakthrough time map.

2.5.1 Facies adaptation using iterative refinement

Adaptation of the grid can be accomplished via the insertion of local grid refinement patches (LGR) at specific locations. Refinement locations can be determined using various techniques, including the use of the cell shape, distance to a well or fault, and flow maps. When using an information map, the criterion we propose for inserting an LGR at a given coarse cell location is to compare some function of the information map over the cell to a given threshold. For instance, using a map of mean local velocities (obtained from a global single-phase flow calculation), if the total flow rate through a particular cell exceeds a given value, an LGR region is introduced at that location. This criterion provides a grid that avoids a large proportion of the total flow going through only a small number of cells. Alternatively, when using a permeability information map, calculating the variance of the underlying fine scale permeability over the coarse cell allows the splitting of cells that span very heterogeneous regions (as in the structured grid approach of Garcia *et al.*, 1992).

This criterion is comparable to an edge-detection technique and can be used to accurately capture the meandering geometry of a channel, as shown in Figure 2.12. A high permeability channel is considered cutting through a low permeability background. The permeability ratio is 100, and the variance of the permeability map is used to construct a series of refined coarse grids (Figure 2.13). To assess the quality of the grids generated, a simple unit-mobility ratio flow simulation with one injector and one producer on each side

Figure 2.12: Two dimensional meandering channel. High-to-low permeability ratio is 100

of the channel is simulated. Flow results (Figures 2.14, 2.15 and 2.16) show that successive refinements improve the match of the total flow rate observed at the producing well in addition to improving the match of the fractional flow curve.

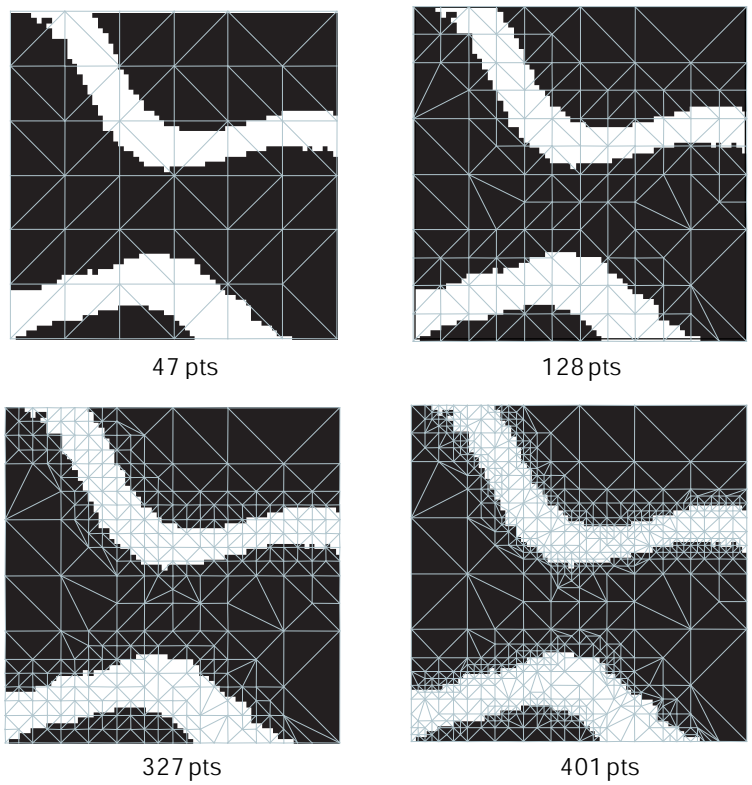


Figure 2.13: Successive triangular grids. The refinement criterion used is the variance of the permeability over the coarse grid cells

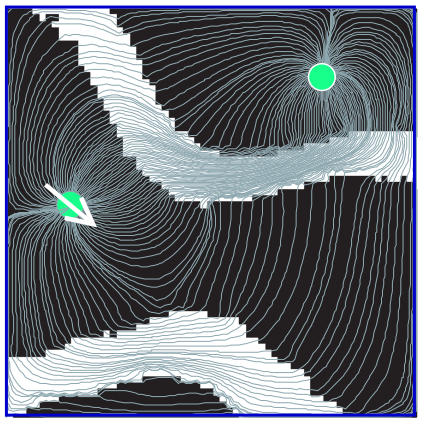


Figure 2.14: Assessment of the quality of the grids in Figure 2.13. Unit mobility-ratio flow simulation with an injector and a producer as shown

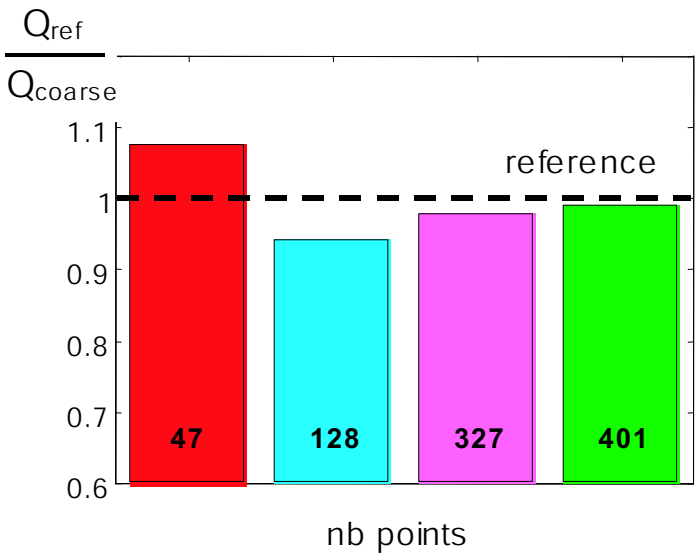


Figure 2.15: Total flow through the coarse model compared to the reference fine scale result

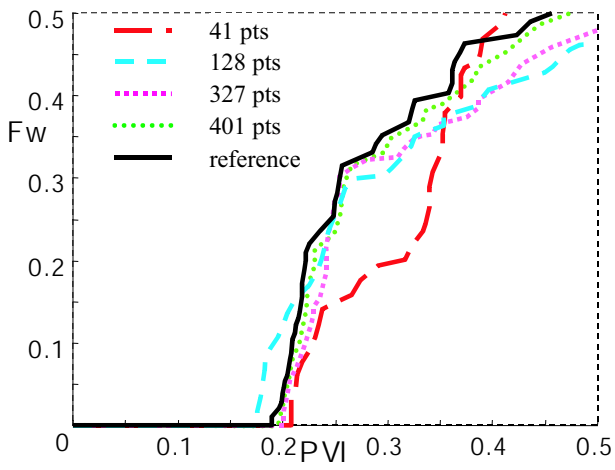


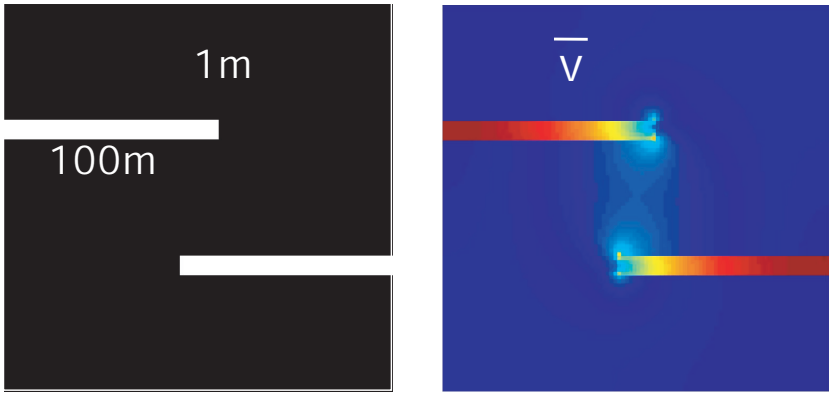
Figure 2.16: Water cut for the reference and coarse models

2.5.2 Adaptation using a target resolution

Gridding can also be performed using *a priori* knowledge of the grid density (as opposed to grids obtained by successive grid refinements). In that case, the information map directly serves as an indicator of the appropriate size of the grid cells at a given location.

Mean velocity map

Using a flow-rate map, a grid can be adapted to the local mean flow intensity under different boundary conditions. Figure 2.17(a) shows two non-connected half layers of high permeability on a low permeability background. Averaging the local velocities for top-to-bottom and left-to-right flow gives the local mean flow-intensity map shown in Figure 2.17(b). The geometry and the rock properties distribution of the problem can be described by a 20×20 Cartesian grid. However, because of numerical errors on such a coarse grid, we will consider the 100×100 refined Cartesian grid as the reference solution. Tracer flow simulations show for instance that water breaks through before 0.5 PVI when using the refined model



(a) (b)

Figure 2.17: High permeability layer incompletely crossing a low permeability zone. A local mean velocity map is obtained when flowing the model top to bottom and left to right

(Figure 2.18) while the coarse Cartesian model does not produce any water for that quantity of pore volume injected (Figure 2.19).

Adapting the triangle sizes to the flow intensity map gives a triangulation (104 nodes) that is naturally denser in high flow regions (Figure 2.20). Comparing the flow results of the adapted coarse grid and a uniform equivalent-size grid to the reference model shows that the adapted grid provides a much more accurate fractional flow curve (Figure 2.21), particularly around breakthrough time. Figures 2.22 and 2.23 compare the water saturation maps at different simulation times for the adapted grid and the uniform triangular grid. The improved characterization of the breakthrough time demonstrates the benefit of the integration of flow information in the gridding process.

Similar techniques can be applied for more realistic reservoir cross-sections, for instance, the layered system shown in Figure 2.24. Those models are particularly challenging for the fractional flow curve reproduction, as thief layers lead to early breakthrough of the water at the producer (see saturation profiles in Figure 2.25). Over-coarsening of these

regions overestimates the breakthrough time, hence making the grid adaptation critical. Using the average flow rate map shown in Figure 2.26, a coarse grid is obtained (Figures 2.27 and 2.28). Flow results for this case are presented in Figure 2.29. Only little improvement of the breakthrough time characterization is achieved by the flow rate adaptation over the equivalent-size uniform grid. Possible reasons for this may be (a) the modeling errors introduced by the well models and (b) the inefficient use of the grid density where too many cells are employed in high flow regions and not enough elsewhere. It is possible that a better target grid density will improve the coarse grid simulations.

Other sources of information can be used to provide the grid size constraint. In the following, we discuss the use of breakthrough time maps.

Breakthrough-time map

A useful quantity provided by the streamline simulation is the time of flight (TOF) of a particle calculated along its path. This TOF serves as a curvilinear coordinate for the resolution of the 1D transport equation along the streamline in the general case of multiphase flow. In the case of tracer flow, the tracer concentration advances as a front at a velocity $\mathbf{u}_f$, related to the total velocity $\mathbf{u}_t$ via.

$$\mathbf{u}_f = \frac{\mathbf{u}_t}{\phi} \tag{2.14}$$

Let τ be the time of flight along a streamline,

$$\tau = \int_0^l \frac{\phi}{|\mathbf{u}_t|}\, dl \tag{2.15}$$

The time of breakthrough of the tracer front along streamline i is then simply equal to the arrival time of the streamline:

$$t^i_{BT} = \tau^i_{\text{end}} \tag{2.16}$$

At breakthrough time, each streamline brings an incremental rate of producing tracer equal to its assigned total flow rate.

If the value of the streamline breakthrough time is assigned along its whole length, this data can be mapped to the 2D grid. We call this map a breakthrough-time map (BT-map). The BT-map allows us to make an immediate correspondence between increments on the producing fractional flow curve and the area of the reservoir that was swept by the incremental tracer produced. In particular, it allows us to identify, for a given configuration of boundary conditions, what is the critical path that needs to be accurately captured by the grid in order to be able to predict the correct breakthrough time using the coarse model. Since the displacement also depends on the velocity, which depends in turn on the pressure field, an accurate description of the area swept by early producing tracer does not *a priori* guarantee that the breakthrough times match. Nevertheless, it turns out to be a useful source of information to guide the refinement.

In the next example, the BT-map is used in the gridding adaptation using a target resolution. Considering the incomplete high permeability streak shown in Figure 2.17(a), fine tracer results can be obtained using streamline simulation. Time of flight contours, corresponding to the tracer concentration front are shown in Figure 2.30. The breakthrough contour map is shown in Figure 2.31 and identifies the region connecting the high permeability layers swept by the tracer at early time in the displacement. We then use the BT-map to compute a grid density map which is used in the metric gridding algorithm (see Borouchaki *et al.*, 1997). The resulting primal grid is shown in Figure 2.32 and can be compared to

Figure 2.20(a). The increased density has shifted from the high permeability layers to the transition zone between the layers. As a consequence, the layer geometry (Figure 2.33) is not as well captured as in the flow rate adapted grid shown in Figure 2.20(b). Flow results (Figure 2.34) exhibit an improvement over the uniform triangular grid. However, the breakthrough time match is not as good as the one obtained with the flow rate adapted grid (Figure 2.21). From the separate use of flow-rate and breakthrough-time information maps, we observed that grid adaptation has a favorable effect on the reproduction of the reference water cut. While the flow rate information leads to an overall better representation of the pressure distribution and therefore of the overall $Q/\Delta P$, the breakthrough time map does not seem to bring as much gain in accuracy as flow rate, if used as a "stand alone" source of information. However, as we will see in Chapter 6, it can be a valuable secondary source of information.

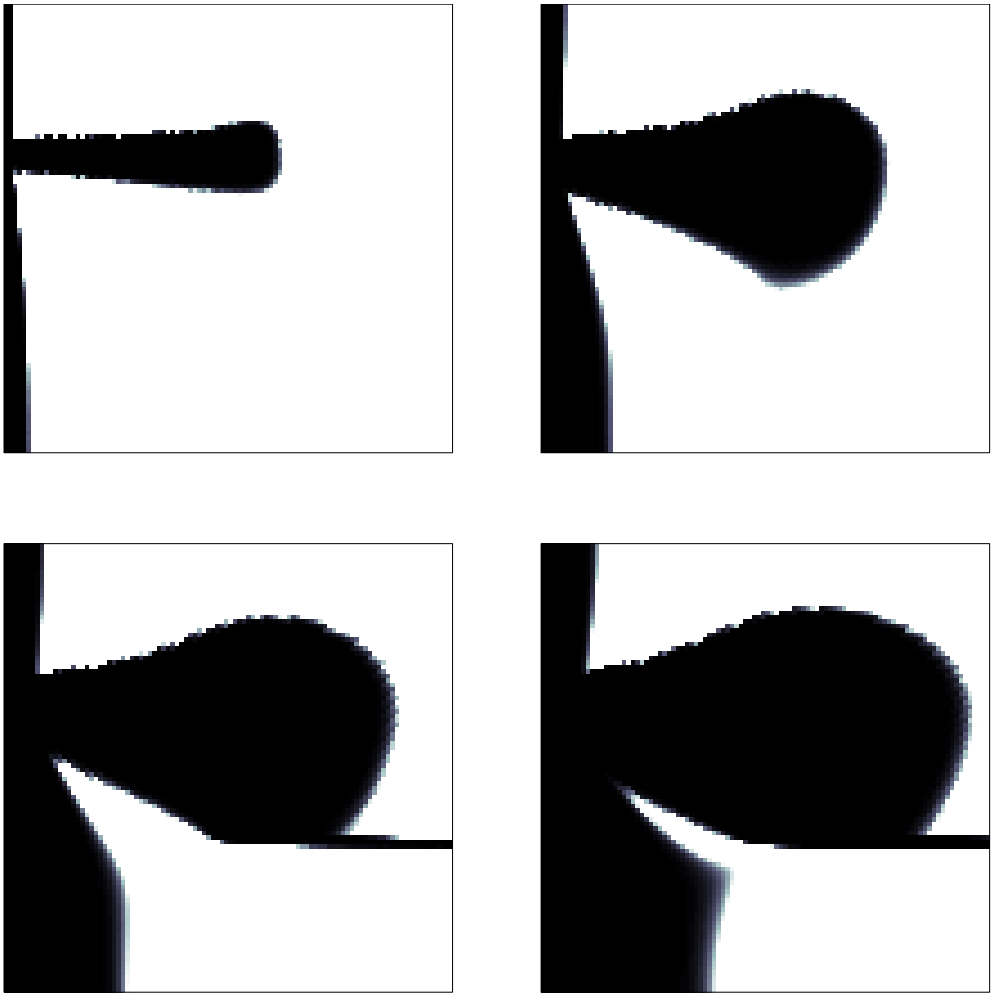


Figure 2.18: Saturation plot for the incomplete layer. Reference solution with 100×100 Cartesian cells. PVI of 0.1, 0.3, 0.5 and 0.7

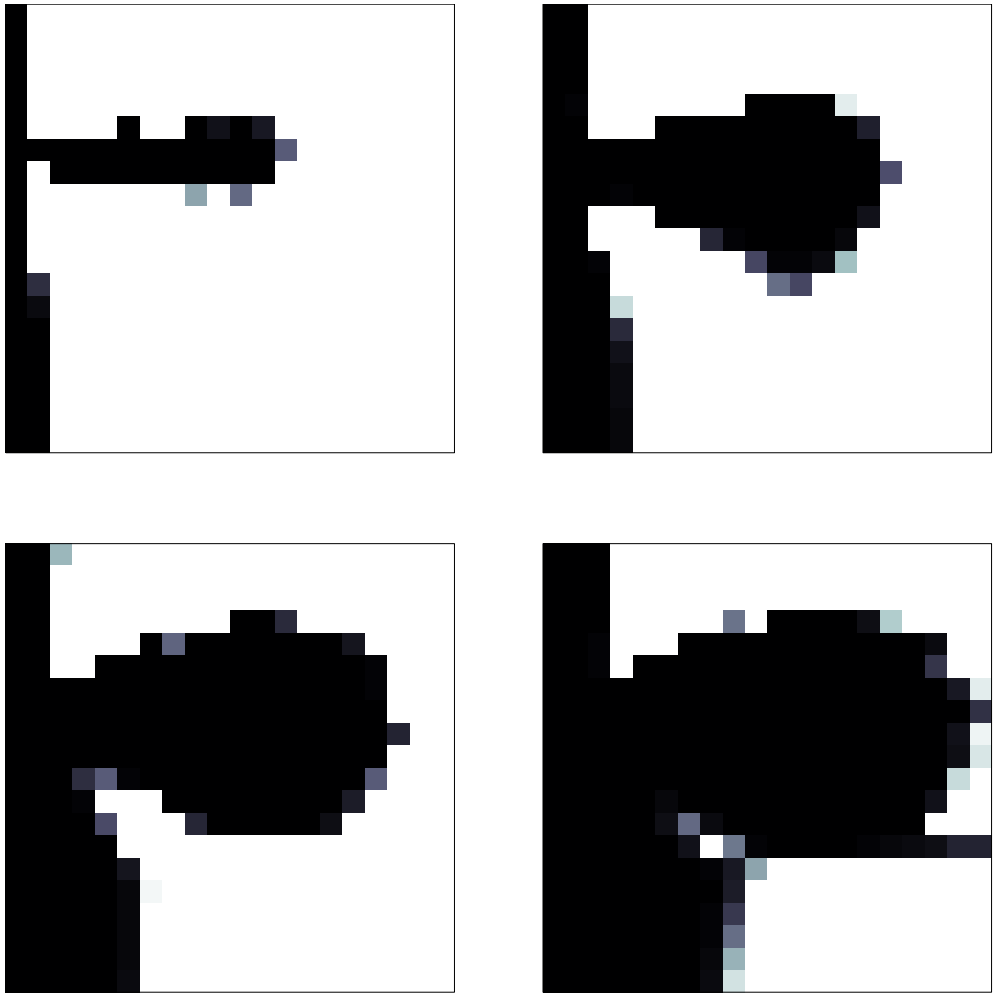
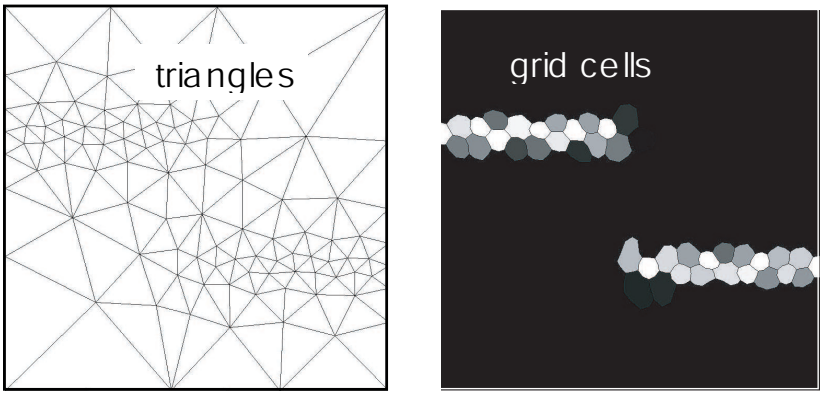


Figure 2.19: Saturation plot for the incomplete layer. Coarse Cartesian solution with 20×20 cells. PVI of 0.1, 0.3, 0.5 and 0.7



(a) (b)

Figure 2.20: Density-adapted coarse model using the mean velocity information. (a) The triangular grid with 104 nodes and (b) corresponding permeability distribution

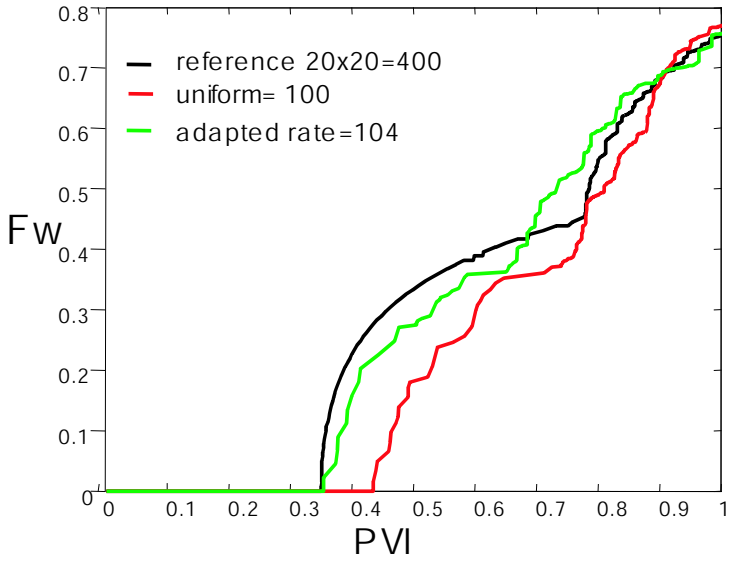


Figure 2.21: For the left to right scenario, water cut is improved over the uniform grid, and breakthrough time is exactly captured

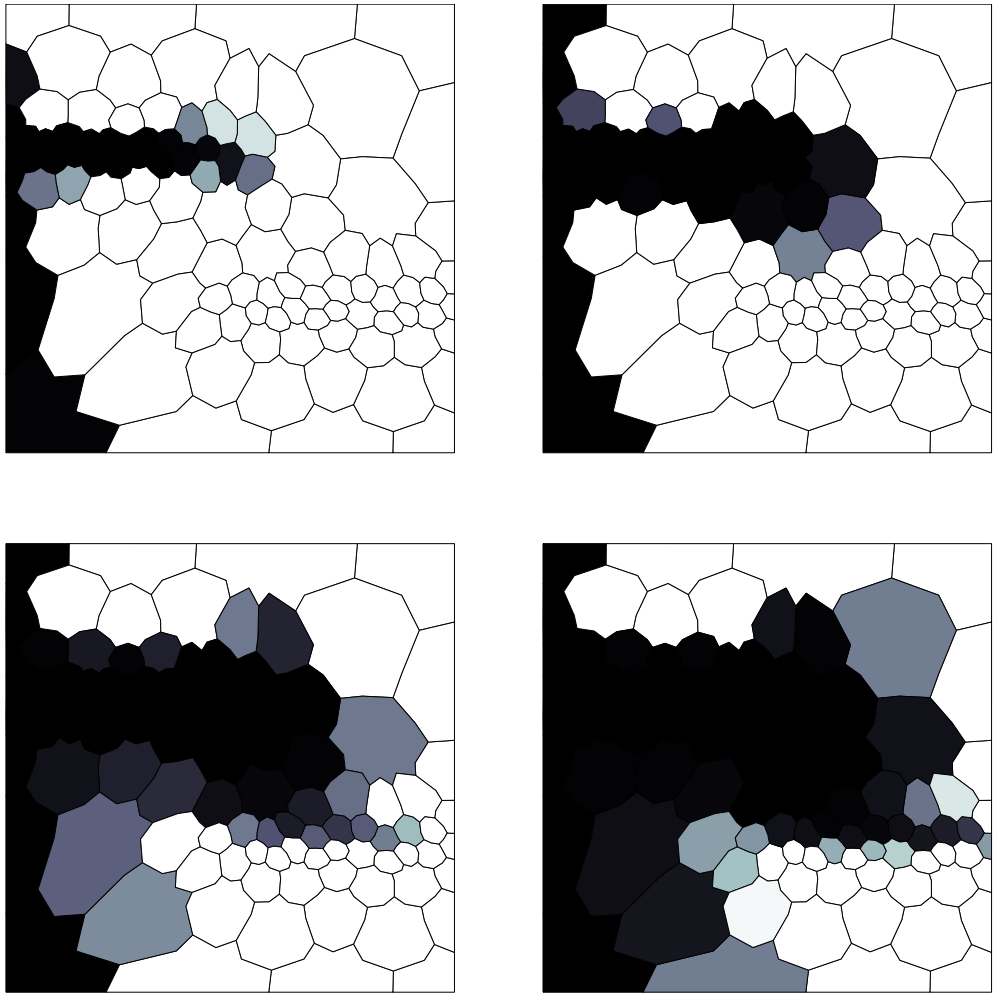


Figure 2.22: Saturation plot for the incomplete layer. Unstructured flow-rate adapted solution with 104 cells. PVI of 0.1, 0.3, 0.5 and 0.7

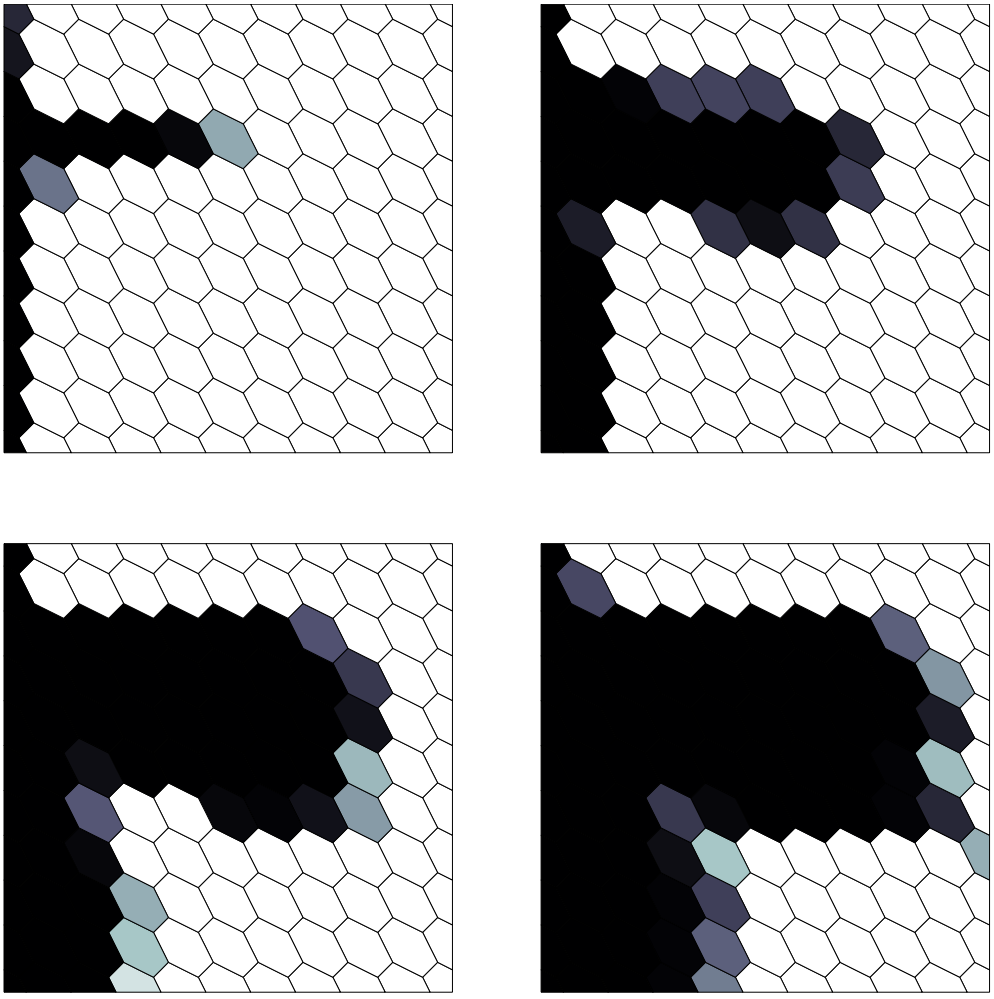


Figure 2.23: Saturation plot for the incomplete layer. Unstructured uniform-grid solution with 100 cells. PVI of 0.1, 0.3, 0.5 and 0.7

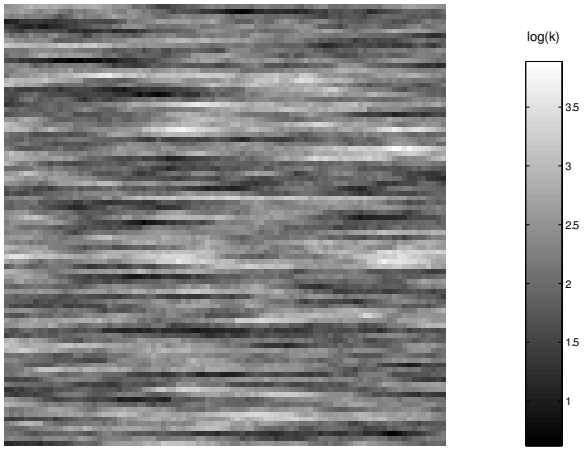


Figure 2.24: Layered system. Permeability field on a log scale

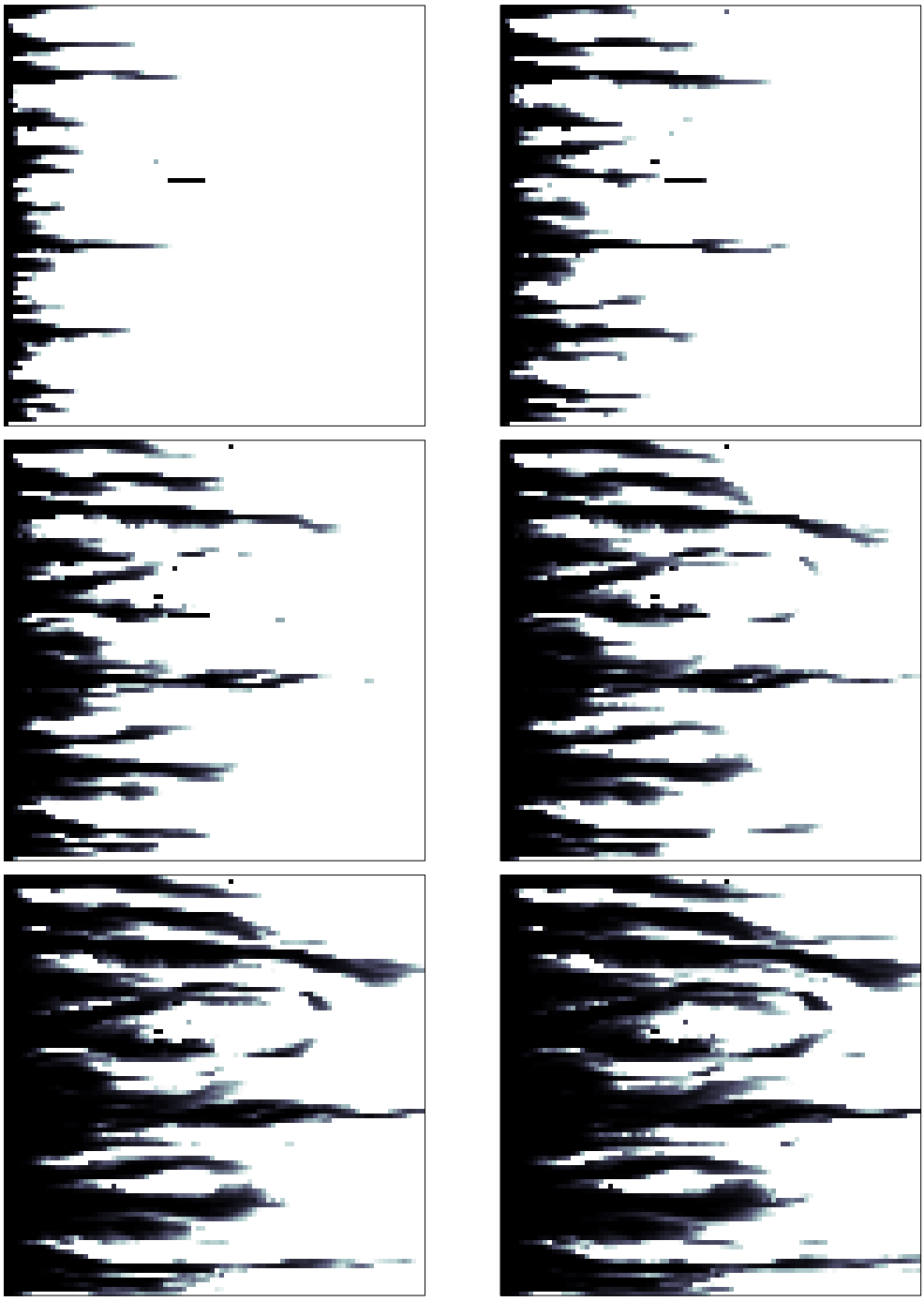


Figure 2.25: Layered system; fine scale saturation maps for PVI of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6. (from left to right and top to bottom)

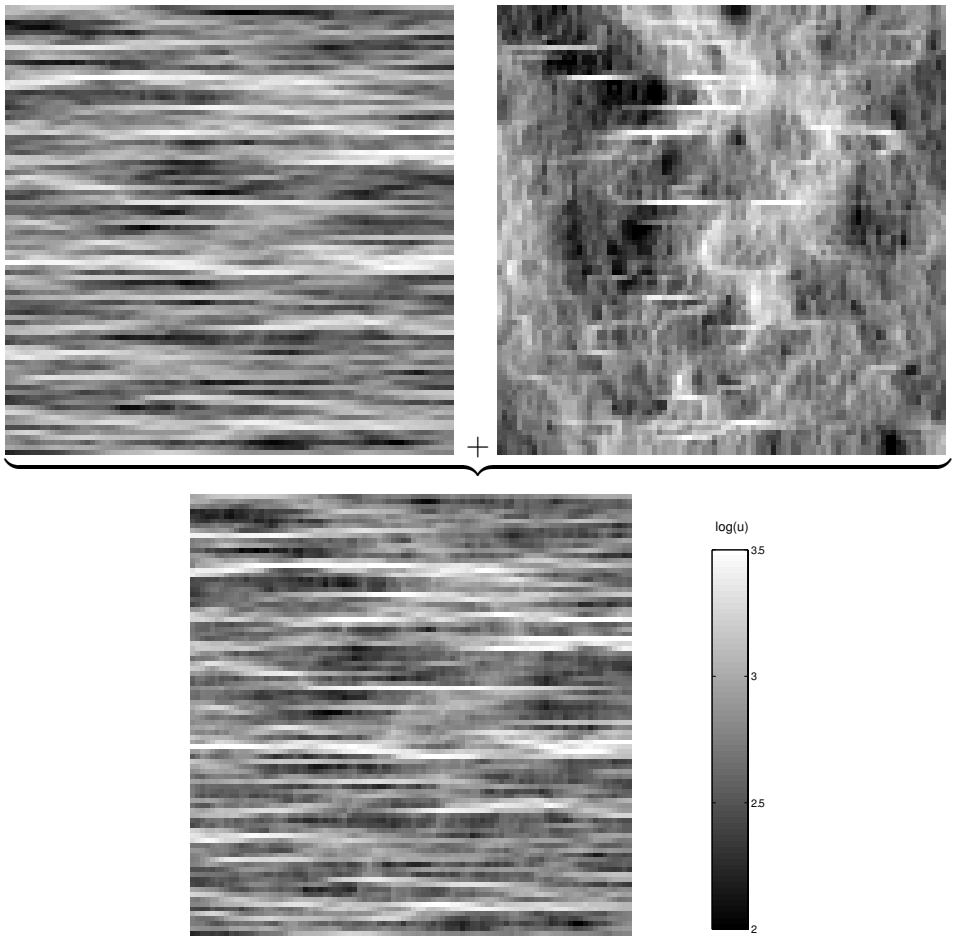


Figure 2.26: The flow rate map is obtained by flowing the model under different boundary conditions and averaging the results (top-left is horizontal flow, top-right is vertical flow)

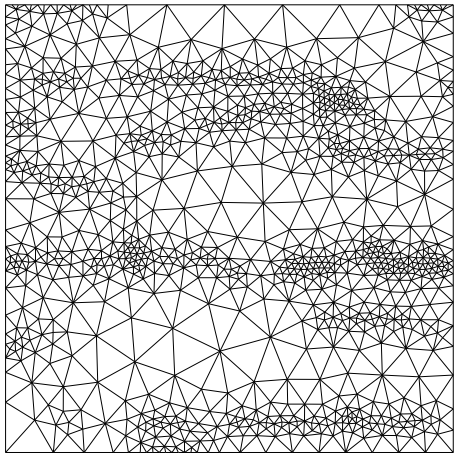


Figure 2.27: Coarse triangular grid with 300 nodes generated from the flow intensity information

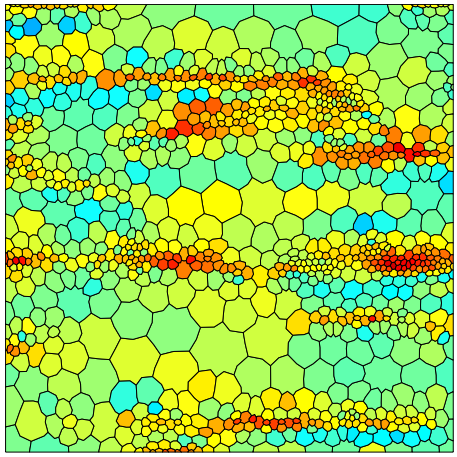


Figure 2.28: Coarse control volume grid generated from the flow intensity information (permeability distribution)

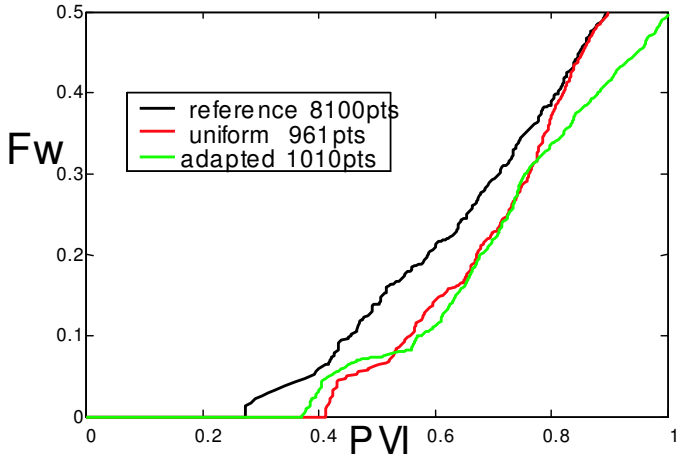


Figure 2.29: Water cut for the flow rate adapted grid shown in Figure 2.28. Marginal improvement is observed for this case

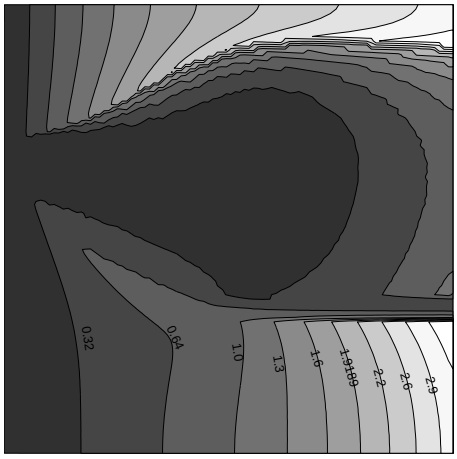


Figure 2.30: Contour of the TOF

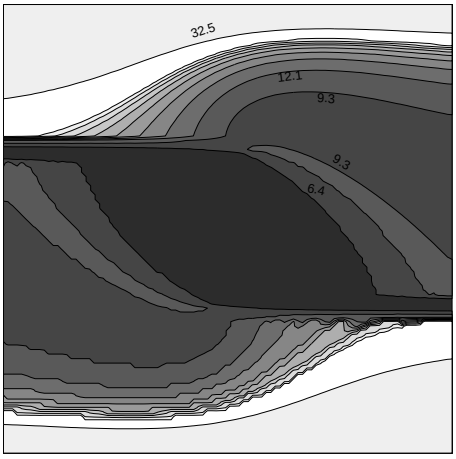


Figure 2.31: Contour of the total TOF (breakthrough map)

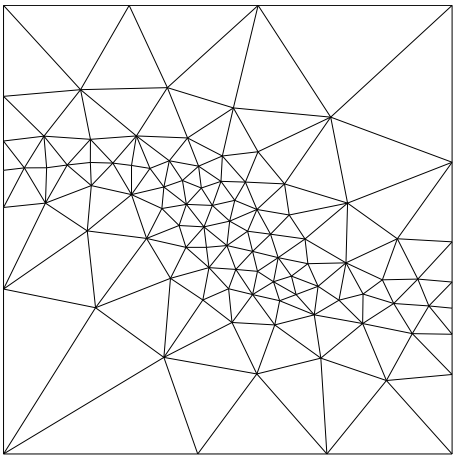


Figure 2.32: Primal grid obtained by TOF adaptation

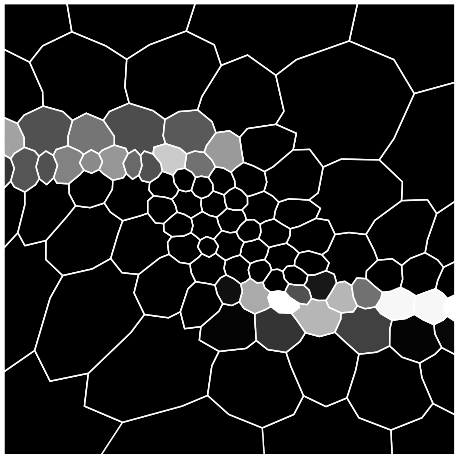


Figure 2.33: Dual grid obtained by TOF adaptation

Calibration of flow information maps

An important aspect of the grid adaptation using information maps is the fact that these maps need to be scaled (or calibrated) to serve as a metric (or a target grid-size). For instance, from a range of average flow-rates obtained on the fine model, different target grid size maps can be inferred. Figure 2.36 shows two grids with equal number of vertices that were obtained using the same flow-rate information map but different calibrations. The correspondence between the information map and the target grid size can be imposed via a calibration curve. The shape of this curve is not known *a priori* and therefore needs to be parameterized in order to allow for some case-specific tuning. An example of a calibration curve is provided in Figure 2.37(a). The chosen parameters are the curve end points and a 'shape factor'. The end points determine the target grid sizes associated with the high and low values of the map while the shape factor relates to the sharpness of the grid size transition between the extreme values. In that respect, the histogram (Figure 2.37(b)) and spatial variability of the map provide valuable information as will be demonstrated in Chapter 6.

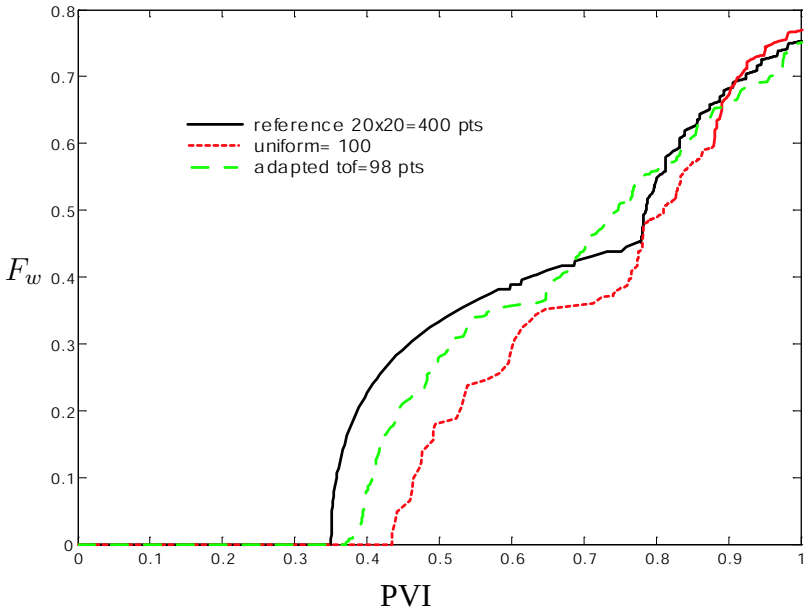


Figure 2.34: Flow result for the TOF adapted grid

2.6 Conclusions for 2D flow-based gridding

The section here above presented a methodology for coarse reservoir model generation that preserves the essential flow characteristics of the fine-scale model. This framework relies on the use of highly flexible unstructured grids, an efficient flow-based method to control the grid quality, and an accurate flow-based permeability tensor upscaling. The coarse grid quality was defined using single phase and tracer flow signatures of the reservoir under different boundary conditions. The two characteristics chosen were (a) the total flow rate across the model and (b) the producing fractional flow curve. It was shown that streamline simulation provides a very effective means to evaluate these characteristics and hence assess the quality of the coarse grid by comparison to the fine model results.

Coarse models generated using grid-density constraints obtained from information maps

exhibited better flow reproduction than equivalent-size non-adapted coarse-models. In particular, the use of local mean flow intensity proved to be effective for capturing early water production in layered reservoirs. The choice of the density map and calibration of the gridding are critical to the adaptation process and to the overall grid quality (for a given number of target coarse cells). Further investigation is needed to create composite criteria for the grid adaptation.

In this chapter, we have seen how to construct 2D adapted models using flow information. We also evaluated the effect of the model coarsening on the flow curves using tracer flow streamline simulation. To make the gridding procedure complete, two additional issues should be addressed: the use of multiple flow scenarios in the flow diagnostic and the calculation of the most effective target grid density from a given flow information map.

Multiple flow scenarios (boundary conditions) may be considered in order to devise flow diagnostics that assess the quality of the coarsening for a variety of flow patterns. This type of flow information may contribute to the robustness of the gridding methodology. Furthermore, the automation of the gridding process is an important aspect of the methodology. Given an information map and a diagnostic tool, the correspondence between information map values and target grid densities must be established. This *calibration* procedure aims at obtaining the most effective grid density from the information map. These issues are further considered in Chapter 6.

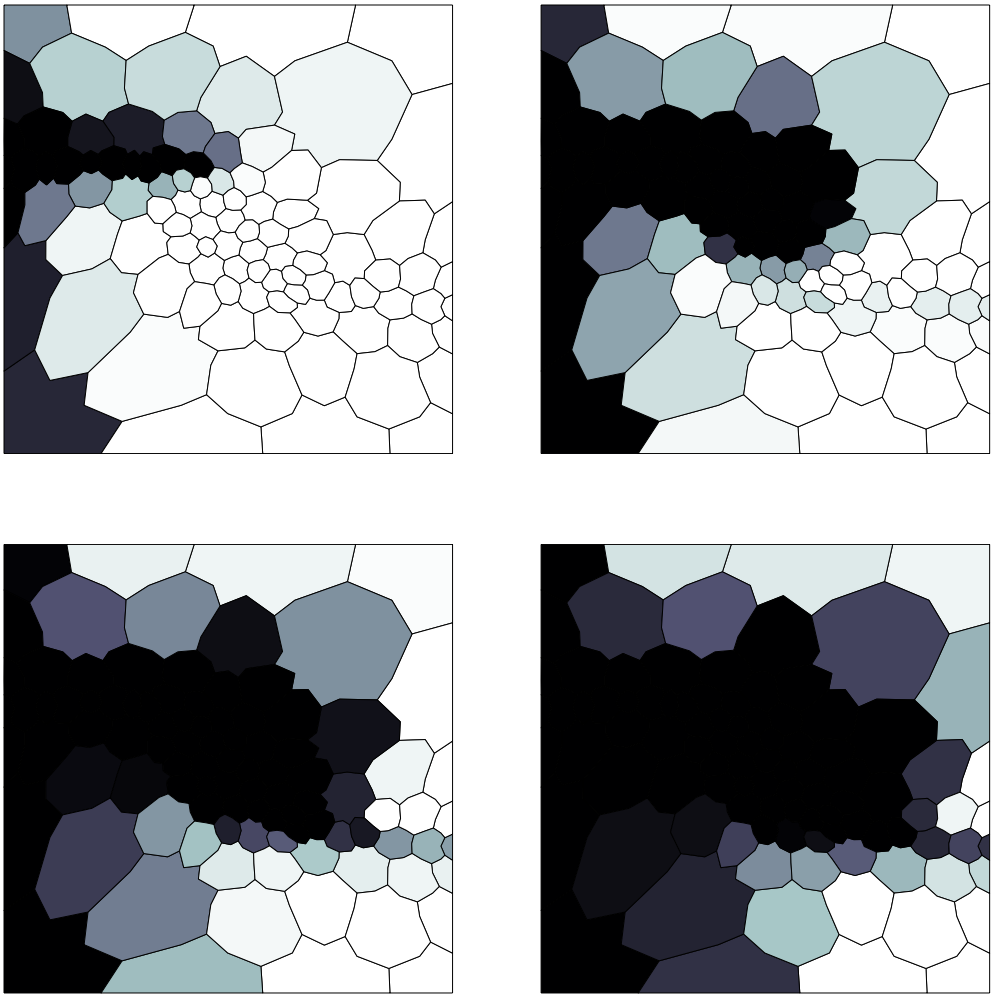


Figure 2.35: Saturation plot for the layered model shown in Figure 2.17. BT-map adapted grid. PVI of 0.1, 0.3, 0.5 and 0.7

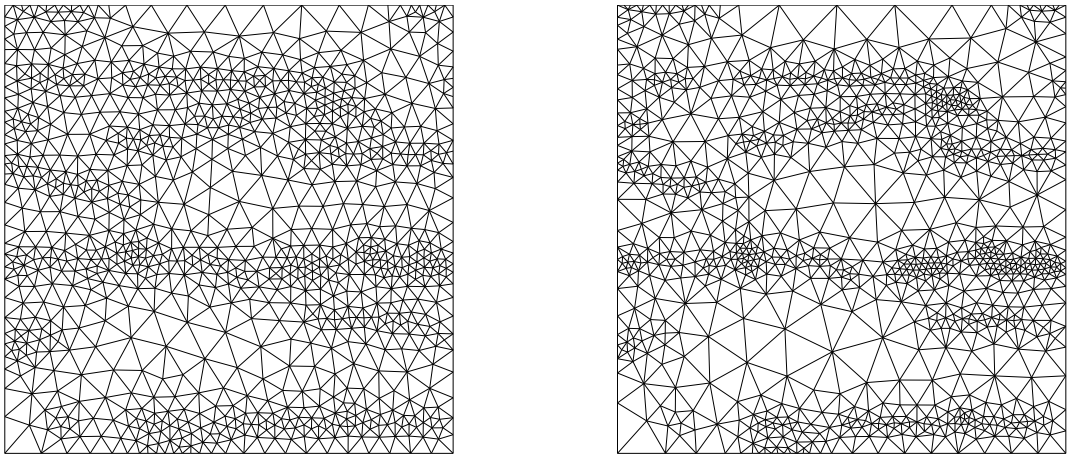
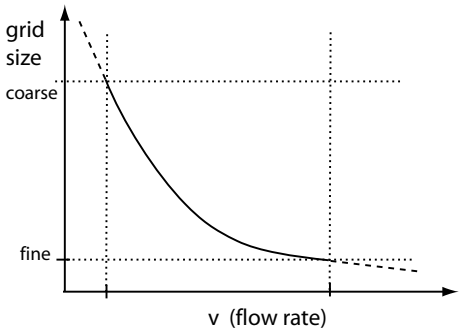
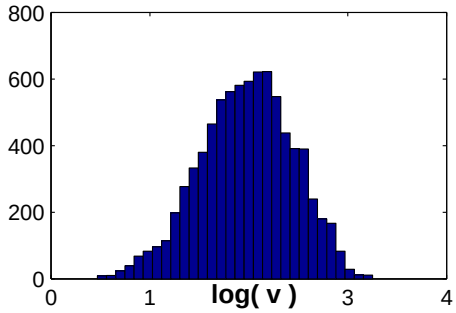


Figure 2.36: Two adapted grids obtained from the same information map (shown in Figure 2.26) but using different calibration. Number of vertices for both grids is 1027



(a) Correspondence between flow rate map and grid size



(b) Histogram of the flow rate map shown in Figure 2.26

Figure 2.37: Calibration of the information map may be achieved via a parameterized curve and considering extreme target grid size values and the information map statistics

Chapter 3

Streamline Simulation on Unstructured Grids

3.1 Introduction

A streamline tracing technique for incompressible flow in porous media on 2D and 3D unstructured grids is presented. The technique entails a local velocity post-processing on a pressure field calculated using a flux-continuous numerical scheme. The reservoirs considered are gridded using a primal grid composed of triangular (2D) or tetrahedral elements (in 3D). The solution of the pressure equation is obtained using a control volume finite element discretization where the pressure values are located at the vertices of the primal grid and the rock properties are assumed to be constant over the dual cells surrounding the pressure nodes. Fluxes recovered from the pressure distribution are then used in a local system to solve for the Darcy velocity on each dual cell. This post-processing involves the introduction of a local sub-grid on which a parameterized form of the velocity is determined using different physical constraints. 2D and 3D techniques are developed, and different degrees

of complexity of the form of the velocity are considered, compared and illustrated.

3.1.1 Previous Work

Accurate schemes for the tracing of particle trajectories is of great importance in modeling transport and multiphase flow in porous media. Two main applications for the tracing of streamlines can be identified: Particle path tracing for visualization purposes, and streamline tracing for displacement modeling problems where one dimensional transport equations are solved on each streamline.

Streamline tracing in visualization

The tracing of a particle path produces a 3D trajectory (streamline) and the curvilinear parameter called time of flight that locates the particle as a function of the curvilinear distance measured on the line. The trajectory often appears as a secondary objective of a numerical method. Hence the choices made to attain the primary objective (which could be the calculation of a potential field for instance), may limit the accuracy of the tracing. Our interest lies in the tracing of streamlines given a pressure field obtained from a lowest order control-volume finite element method (CVFE). This method provides continuous fluxes but it is not primarily devised to provide accurate velocities.

Several authors have proposed post-processing methods that operate on discrete (discontinuous) velocity fields to provide an improved tracing (see Cordes and Kinzelbach, 1992; Pokrajac and Lazic, 2002). However, these methods essentially concerned velocities derived from finite element (FE) methods. Since FE methods are not locally conservative over the primary grid (though mass conservation can be recovered; see Hughes *et al.*, 2000), the proposed post-processing method must include a component of conservation of mass.

This was not done in Cordes and Kinzelbach (1992) or Pokrajac and Lazic (2002). Therefore, the flow-rates assigned to the streamlines are not well defined, making the streamlines unsuitable for the purpose of transport modeling.

Cordes and Kinzelbach (1992) proposed a post-processing involving the calculation of a piecewise constant velocity field on a refined grid. Their method gave satisfactory tracing in two dimensions for velocities obtained using FE methods but as the authors' intent was primarily to exhibit meaningful tracing, they did not provide any displacement results. Furthermore, in later work, Cordes and Kinzelbach (1994) state that it is not possible to obtain post-processed velocities in 3D that satisfy both $\nabla \cdot \mathbf{u} = 0$ and $\nabla \times \mathbf{k}^{-1} \nabla P = 0$. In particular, they argue that violating the second constraint leads to non-physical "swirling" of the streamlines. This behavior was even observed for Pollock tracing on Cartesian grids. In Section 3.3, the difficulty in obtaining an irrotational velocity field is acknowledged but a solution to minimize this effect is proposed.

Streamline method

For applications that involve the mapping of a transport equation onto the pathlines as in the streamline method (Batycky *et al.*, 1997), the computation of accurate trajectories is critical. The nature of the streamline method permits a 1D semianalytical treatment along streamlines for fluid transport, which minimizes dispersivity effects. As a consequence, the transport solution strongly relies on the accurate computation of both streamline trajectories and time of flight, which serves as a curvilinear coordinate for the transport equation. In 2D, the analogy between the streamline and the streamtube (Thiele, 1994) methods highlights the need for accurate rate assignment to the streamlines. Hence, conservative numerical schemes are generally preferred for the discretization of the pressure equation. For

unstructured grids, flux-continuous control-volume finite element methods (CVFE) have successfully been applied to the resolution of pressure using permeability that is piecewise constant over the primal cells (Forsyth, 1990) or piecewise constant over the dual cells or control volumes (Verma and Aziz, 1997), as described in Chapter 2.

An important family of flux-continuous methods was developed by Verma and Aziz (1997), Aavatsmark *et al.* (1998), and Edwards (2002) for CVFE methods. Their transmissibility calculations generalize the flux-continuous permeability harmonic mean used for Cartesian grids to unstructured grids. From now on, we consider the case of a pressure field obtained on an unstructured grid using a CVFE method with the aforementioned flux continuous schemes. Two distinct permeability approximations (cell-centered or point distributed; Figure 3.1) can be considered in the CVFE, leading to different discrete models. The point-distributed approximation (Settari and Aziz, 1972) is the most commonly applied approximation in reservoir simulation. Verma (1996) demonstrated that it also leads to more accurate results. The Cordes and Kinzelbach (1992) patch recovery (initially devised for a Galerkin FE method) is immediately applicable to the CVFE method in the case of a cell-centered permeability approximation (in which case the CVFE and Galerkin methods provide identical pressure fields for the incompressible problem). Prévost *et al.* (2002) extended the recovery method to the point-distributed permeability approximation. The tracing method entails the use of an isoparametric mapping, followed by a bilinear tracing in a reference domain.

3.1.2 Objective

The present chapter summarizes the velocity post-processing technique in 2D and its extension to 3D. Emphasis is put on the construction of a system of constraints that is physically

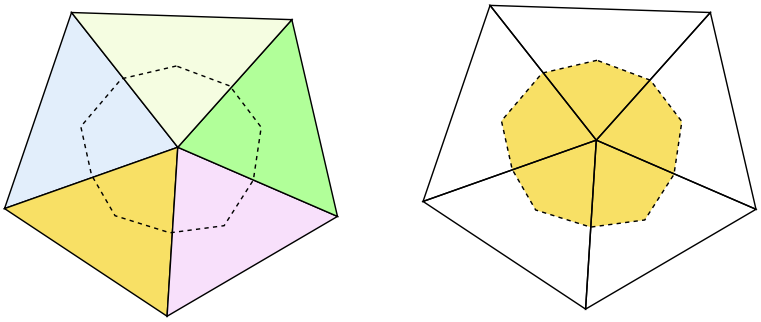


Figure 3.1: Cell centered (left) and point distributed (right) permeability approximations. Primal grid in solid line, dual grid in dashed line

meaningful and that is equal in size to the number of parameters defining the target velocity interpolation. A discussion of possible system closure conditions is also included.

3.2 Two dimensional tracing

As indicated above, accurate streamline tracing and streamline simulation were obtained by Prévost (2000) and Prévost *et al.* (2002) for CVFE pressure discretizations. Most of the 2D formulation and results were presented by Prévost (2000) in MS work. This is included here for completeness and to better elucidate the 3D case. Two tracing techniques were presented, both based on a patch recovery technique as suggested by Cordes and Kinzelbach (1992). This section summarizes the two tracing techniques, one involving a bilinear approximation of the velocity on an unstructured sub-quadrilateral grid, the other involving a piecewise constant velocity approximation on a sub-triangular grid.

3.2.1 Notation

The following table describes the notations used in the present section.

Notation	Definition
Roman	
N_f	Total number of triangular faces in the gridded domain
n_f	Number of faces connected to a node
n_e	Number of edges connected to a node
$\mathbf{V}^\circ$	Natural velocity for the region at the intersection of the patch p and the triangle f
$\mathbf{V}^*$	Recovered velocity for the region at the intersection of the patch p and the triangle f
$\mathcal{F}$	Triangle (face)
$\mathcal{Q}$	Quadrilateral
$\mathcal{E}$	Edge
$\mathcal{M}$	Midpoint of edge
$\mathcal{C}$	Barycenter of a face
p	Vertex of the primal grid
$\hat{\mathbf{n}}$	Normal vector
$\hat{\mathbf{t}}$	Tangential vector
f	Flux across an edge
Superscript	
p	Relative to a patch or equivalently to a node p
f	Relative to a triangle

continued on next page

Notation	Definition
$\circ$	Refers to a quantity obtained directly from the numerical approximation (without any post-processing)
$*$	Refers to post-processed quantities
Subscript	
m	Index of a triangle within a patch
i	Interior sub-triangle (always relative to a patch and a triangle)
e	Exterior sub-triangle (always relative to a patch and a triangle)
a, b	Adjacent quadrilaterals or triangles
x, y	Physical space coordinates
ξ, η	Reference space coordinates

3.2.2 Grids and numerical schemes

A common choice for the locations of the pressure unknowns are the vertices of the unstructured grid. Alternate choices such as the center of the triangle or the edge midpoints (in 2D) will not be discussed. The grid connecting the pressure nodes will be referred to as the primal grid. As mentioned above, the permeability can be chosen to be piecewise constant either over the cells (triangles) or on the control volume surrounding the pressure node (point distributed). From now on, we consider the case of point distributed permeabilities for the CVFE method (Figure 3.2).

Let us consider the discretization of the pressure equation on a volume V in integral

form:

$$\oint_{\Gamma_V} \Lambda \mathbf{k} \nabla P \cdot \hat{\mathbf{n}} \, ds = 0 \tag{3.1}$$

where Γ_V denotes the perimeter of V.

Without loss of generality, we can uniquely define the flux on the segments of Γ_V:

$$f_j^m \equiv \int_{\mathcal{E}_j^m} \Lambda \mathbf{k} \nabla P \cdot \hat{\mathbf{n}} \, ds \qquad \text{with} \quad m \in \{1, \cdots, n_f\} \quad \text{and} \quad j \in \{1, 2\} \tag{3.2}$$

where $\mathcal{E}_j^m$ is the jth segment of the control volume that belongs to the triangle m (see also Figure 3.3).

The question of how the fluxes are approximated in Equation 3.2 leads to a variety of flux-continuous schemes (Fung and Nghiem, 1990; Palagi and Aziz, 1991; Verma and Aziz, 1997; Aavatsmark *et al.*, 1998; Edwards, 2002). Verma and Aziz (1997) proposed an approximation of the flux as a linear combination of pressures that preserves the flux continuity in the case of control volume distributed properties. In this approximation (valid for a general full permeability tensor), the pressure is assumed to vary linearly within each quadrilateral composing the control volume (see Section 2.3.2 for details).

3.2.3 Natural velocity

Control-volumes can be sub-divided into n_f quadrilaterals (Figures 3.2 and 3.3). In the CVFE discretization, the fluxes on the perimeter of a control volume p are obtained assuming a linear pressure within each quadrilateral $\mathcal{Q}$ between the nodes p, $\mathcal{M}_1$ and $\mathcal{M}_2$ as shown in Figure 3.4. Since the permeability is assumed constant over the control volume, it follows that a Darcy velocity $\mathbf{V}^\circ$ can be obtained for the quadrilateral such that it produces

the same flux across the control volume edges (as defined in Figures 3.3 and 3.5):

$$\begin{cases} \mathbf{V}^\circ \cdot \hat{\mathbf{n}}_1 &= f_1^\circ / |\mathcal{E}_1| \\ \mathbf{V}^\circ \cdot \hat{\mathbf{n}}_2 &= f_2^\circ / |\mathcal{E}_2| \end{cases} \tag{3.3}$$

This velocity is called the "natural velocity" because it directly derives from the linear pressure assumption in the CVFE method (see also Figure 3.6).

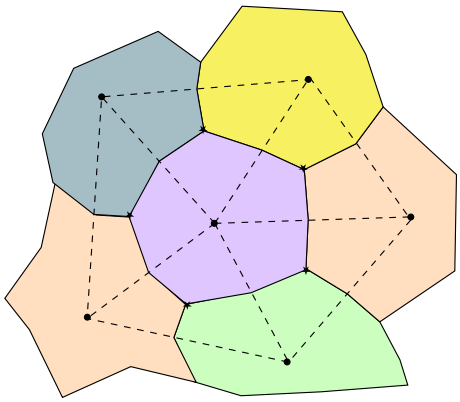


Figure 3.2: Control volume

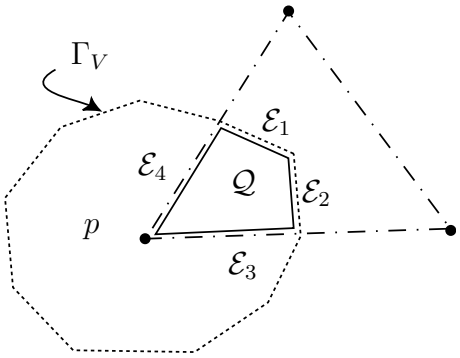


Figure 3.3: Definition of the edges of $\mathcal{Q}$

This natural velocity is, however, not satisfactory for tracing streamlines. In particular, the normal component of the velocity is generally discontinuous across the edges $\mathcal{E}_3$ and $\mathcal{E}_4$

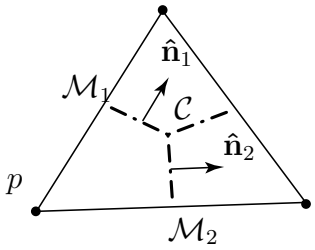


Figure 3.4: Definition of the points of the control volume and normal vectors $\hat{\mathbf{n}}_1$, $\hat{\mathbf{n}}_2$

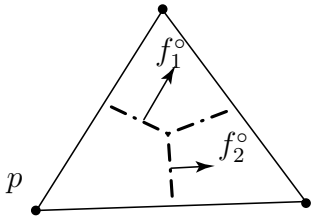


Figure 3.5: Fluxes directly obtained from the numerical method

of the quadrilateral. This discontinuity occurs because the flux continuity is only enforced on $\mathcal{E}_1$ and $\mathcal{E}_2$ in the numerical scheme. This discontinuity results in non-conservation of mass along the streamlines and incorrect tracing (see Figures 3.7 and 3.8) if $\mathbf{V}^\circ$ is used directly.

3.2.4 Criteria for post-processed velocity

Three criteria are defined for the velocity field $\mathbf{V}$ in order to achieve an accurate streamline tracing:

1. The velocity field must satisfy $\nabla \cdot \mathbf{V} = 0$ at all points.

2. Where $\mathbf{V}$ is not differentiable, the normal component of $\mathbf{V}$ must be continuous across each discontinuity, ensuring mass conservation along the streamline.

3. On each segment Γ_j of a control volume, $\mathbf{V}$ must be consistent with the flux f° given

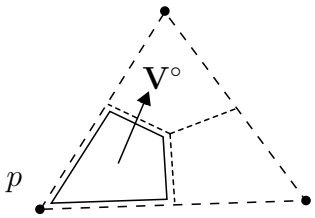


Figure 3.6: Natural velocity

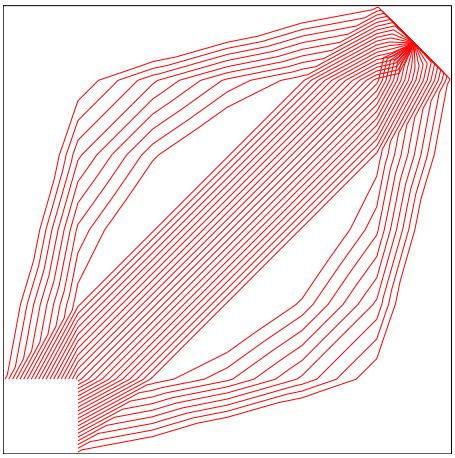


Figure 3.7: Using the natural velocity to perform the tracing leads to incorrect results

by the solution of the pressure equation:

$$\int_{\Gamma_j} \mathbf{V} \cdot \hat{\mathbf{n}}_j \, dl = f_j^\circ \tag{3.4}$$

In addition, we note that Darcy's law implies that for any subdomain γ over which permeability is homogeneous, we have:

$$\int_\gamma \nabla \times \mathbf{V} \, ds = -\mathbf{k} \int_\gamma \nabla \times \nabla P \, ds = 0 \tag{3.5}$$

This property requires that the flow is locally irrotational (*i.e.*, over regions of constant permeability).

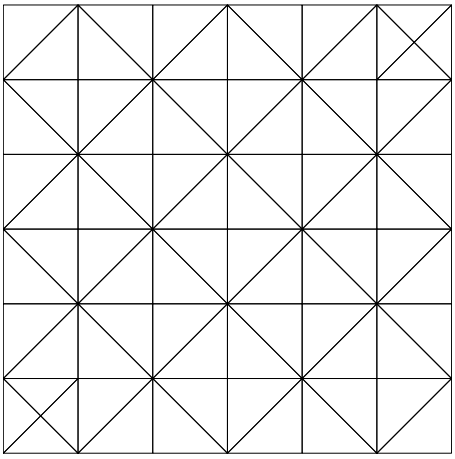


Figure 3.8: Uniform triangular grid use to solve for the pressure on Figure 3.7

3.2.5 Recovered velocity

This section addresses the numerical determination of the unknown velocity, referred to as $\mathbf{V}^*$, subject to the tracing-quality constraints defined in 3.2.4. Two possible parameterized forms for $\mathbf{V}^*$ are considered and a linear system enforcing the constraints is devised. Finally, closure of the system is discussed followed by comments on the uniqueness of such a solution.

Two possible approximations

The problem of finding an expression for the velocity inside the control volume that also satisfies the quality criteria defined in Section 3.2.4 is considered here. In the approach that we propose, the velocity $\mathbf{V}^*$ is parameterized on a sub-grid filling the control volume. This sub-grid is called a patch. At least two approximations (and their corresponding patch division) can be identified:

- a bilinear interpolation based on the quadrilateral defining the control volume,

- a piecewise constant approximation based on sub-triangles obtained by splitting the aforementioned quadrilateral.

In the second case, the patch is a sub-triangular grid composed of $2n_f$ elements if p is not a boundary node (Figure 3.9).

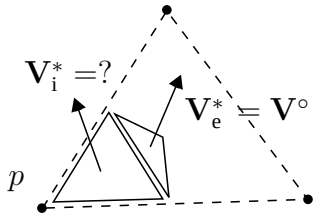


Figure 3.9: Consistency constraints completely determine the velocity in the external sub-triangle

Consistency and continuity conditions

Given a choice for the form of the velocity (piecewise constant or piecewise linear) and the corresponding sub-division of space (patch), the consistency and continuity constraints are written respectively at the perimeter of the patch and at its interior boundaries.

In the case of a bilinear interpolation, there is one quadrilateral element per triangle connected to p. Each parameterized velocity $\mathbf{V}^*$ is entirely defined by the values of the four fluxes on the boundaries f_j^* where $j \in \{1, 2, 3, 4\}$ and the geometry of the quadrilateral (see Figure 3.10). Applying directly the consistency constraints, we immediately see that f_j^* must be equal to f_j° for $j = 1$ and $j = 2$. A divergence-free expression is achieved only if we also require:

$$\sum_{j=1}^{4} f_j^* = f_1^* + f_2^* + f_3^* + f_4^* = 0 \tag{3.6}$$

This is equivalent to

$$f_3^* + f_4^* = -\left(f_1^\circ + f_2^\circ\right) \tag{3.7}$$

This condition is necessary to obtain a divergence-free expression for $\mathbf{V}^*$, though it is not sufficient. Prévost *et al.* (2001) showed that the use of an isoparametric mapping transforms a divergence free expression for the velocity in reference space into a velocity in physical space that approximates zero-divergence to first order.

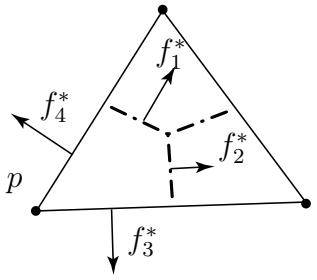


Figure 3.10: Location of the fluxes to be obtained by post-processing

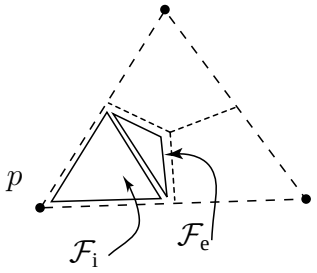


Figure 3.11: A sub-patch is composed of 2 sub-triangles, the interior ($\mathcal{F}_\mathrm{i}$) and the exterior ($\mathcal{F}_\mathrm{e}$)

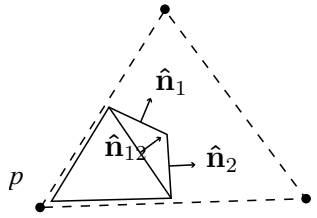


Figure 3.12: Normal vectors

The continuity constraints are enforced at each of the n_e interior boundaries of the patch. For the two adjacent quadrilaterals $\mathcal{Q}_a$ and $\mathcal{Q}_b$, we have (see also Figure 3.13):

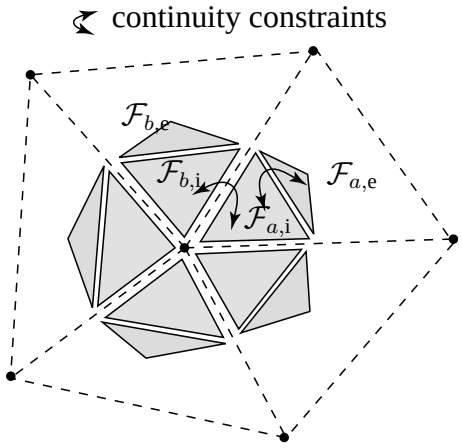


Figure 3.13: Continuity constraints between exterior and interior sub-triangles

$$f^*_{a,3} = -f^*_{b,4} \tag{3.8}$$

For any node p in the interior of the gridded domain, we have the important relation,

$$n_e = n_f \tag{3.9}$$

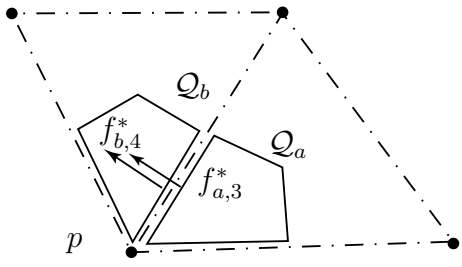


Figure 3.14: Continuity conditions between quadrilaterals a and b

which guarantees that there are as many linear constraints as degrees of freedom for the expression of the velocity within the patch.

System closure

By summing all the equations obtained in Equations 3.7 and 3.8 separately, we gather:

$$\begin{cases} \sum_{m=1}^{n_f} f^*_{m,3} + \sum_{m=1}^{n_f} f^*_{m,4} &= -\sum_{m=1}^{n_f} \left(f^\circ_{m,1} + f^\circ_{m,2} \right) \\ \sum_{m=1}^{n_f} f^*_{m,3} + \sum_{m=1}^{n_f} f^*_{m,4} &= 0 \end{cases} \tag{3.10}$$

As a consequence, the rank of the linear system is at best $n_f - 1$ (independent of the values of the fluxes imposed on the boundary). We can identify two possible cases:

- the net flux vanishes, in which case there exists an infinite number of solutions,
- the net flux does not vanish, in which case there is no solution satisfying all conditions and a different interpolant form must be employed for the tracing.

For the case of an incompressible flow, away from sinks and sources, we always have an infinite number of solutions. In the general case (away from domain boundaries and sources), the deficiency in the rank can be cured by adding one extra equation to the system. Cordes and Kinzelbach (1992) suggested the use of a linear equation enforcing the curl of the velocity field to be zero in a weak sense:

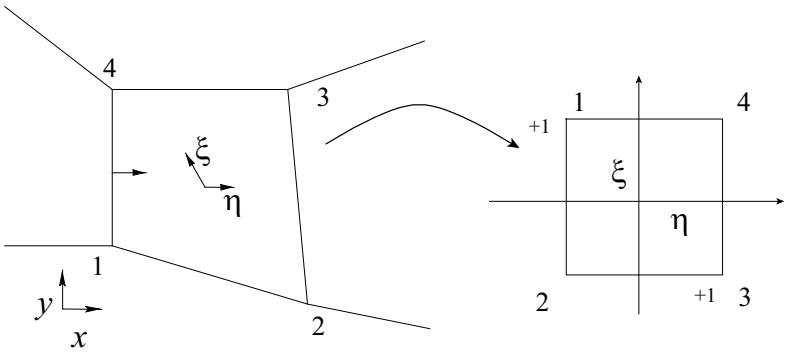


Figure 3.15: Isoparametric transform

$$\begin{aligned} \iint_{\Omega} \nabla \times \nabla P \, ds &= \iint_{\Omega} \nabla \times \mathbf{k}^{-1}\mathbf{V} \, ds \\ &= \mathbf{k}^{-1} \oint_{\Gamma} \mathbf{V} \cdot \hat{\mathbf{t}} \, dl \\ &= 0 \end{aligned} \tag{3.11}$$

with $\Omega = \bigcup_m \mathcal{F}^p_{m,\mathrm{i}}$ the union of all the interior sub-triangles in the patch p. This equation provides closure for the system and uniqueness for the solution of recovered velocities.

The problem that arises when using a sub-quadrilateral grid for the tracing is that the analytical form of $\mathbf{V}^*$ on each sub-quadrilateral is only explicitly known in the unit square. The path-line on which the circulation has to be calculated is composed of the outer edges of the quadrilaterals. Mapping to a unit cell in (ξ, η) reference space gives:

$$\int_{\mathcal{E}_i} \mathbf{V}^*(x,y) \cdot \hat{\mathbf{t}} \, dl = \int \mathbf{V}^*(\xi,\eta) \cdot \hat{\mathbf{t}} \; |J| \; dl \tag{3.12}$$

where J is the Jacobian of the isoparametric transformation of a point (ξ, η) in reference space into a point in physical space (x, y), as shown in Figure 3.15.

To avoid approximating Equation 3.11 in physical space, we choose a simpler solution which consists in using a piecewise constant form of the velocity and analytically calculate its circulation. Doing this, we avoid the calculation of J and a numerical integration along the edge, or the possible errors introduced by the assumption $J(\xi, \eta) = \text{constant}$. The drawback of this choice is that we need to introduce a new sub-division of the patch and write the linear system expressing the velocity solution in terms of the new parameters. Indeed, we need to rewrite the system of constraints from a formulation where the unknowns are on the edges of the quadrilaterals to a formulation where the unknowns are the two components of the velocity within sub-triangles. This technique is almost identical to the one explained by Cordes and Kinzelbach (1992) with the exception that it is also applicable to point distributed grids.

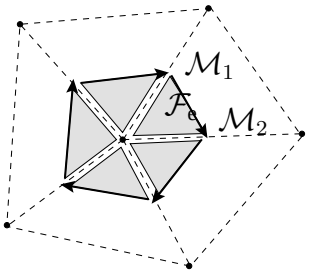


Figure 3.16: Closure of the system is obtained by setting the circulation of the piecewise constant velocities to zero on the interior sub-triangles

Tracing on sub-triangles

Each quadrilateral of the patch is divided into two sub-triangles, one connected to the node p (the interior sub-triangle $\mathcal{F}_\text{i}$) and one not connected to p (the exterior sub-triangle $\mathcal{F}_\text{e}$)

as shown in Figure 3.11. The velocity is assumed to be piecewise constant on the sub-triangular grid composing the patch.

In the same fashion as for the quadrilateral case, the constraints are written for this velocity expression.

- At each point not on any edge of the sub-triangular grid, we have $\nabla \cdot \mathbf{V}^* = 0$
- The consistency equations are written on the exterior sub-triangles only (two constraints per $\mathcal{F}_{\text{e}}$) and hence identify the post-processed velocity as the natural velocity (Equation 3.13 and Figure 3.9).

$$\mathbf{V}^*_{\text{e}} = \mathbf{V}^\circ \tag{3.13}$$

- The only constraints that remain to be enforced are continuity constraints among adjacent interior sub-triangles and continuity constraints between interior and exterior sub-triangles of the same sub-patch (see also Figure 3.13),

$$\begin{cases} \mathbf{V}^*_{m,\text{i}} \cdot \hat{\mathbf{n}}^m_{12} &= \mathbf{V}^*_{m,\text{e}} \cdot \hat{\mathbf{n}}^m_{12} \\ \mathbf{V}^*_{a,\text{i}} \cdot \hat{\mathbf{n}}^a_1 &= \mathbf{V}^*_{b,\text{i}} \cdot \hat{\mathbf{n}}^a_1 \\ &= -\mathbf{V}^*_{b,\text{i}} \cdot \hat{\mathbf{n}}^b_2 \end{cases} \tag{3.14}$$

The same rank deficiency in the system as for the quadrilateral case is observed but now the closure condition can readily be written,

$$\sum_{m=1}^{n_f} \mathbf{V}^*_{m,i} \cdot \mathbf{x}^m_{12} = 0 \tag{3.15}$$

where $\mathbf{x}^m_{12}$ is the vector pointing from $\mathcal{M}_1$ to $\mathcal{M}_2$ in the triangle m attached to p

(Figure 3.4).

Once the full-ranked system of equations is solved, the tracing can be performed on the sub-triangular grid or the sub-quadrilateral grid.

Note that an alternate subdivision of the patch into triangles may be considered as shown in Figure 3.17. This grid is obtained by splitting the quadrilateral composing the control volume along the diagonal that contains the center of the patch. The post-processing described above may be applied to this alternate velocity discretization. Similar subdivision is used in the 3D case. It is discussed in Section 3.3.

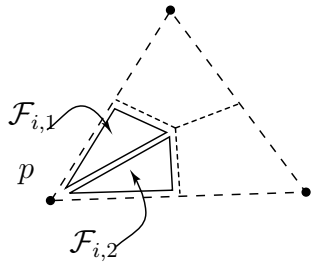
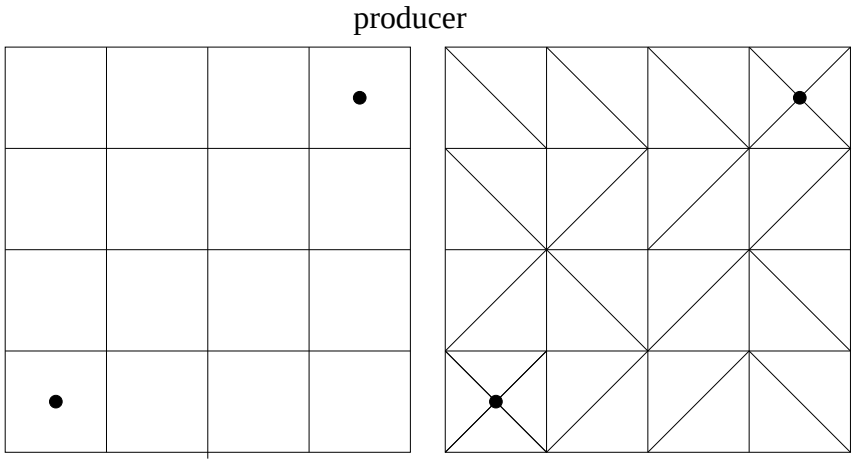


Figure 3.17: Alternate construction of the sub-patch

Tracing on sub-quadrilaterals

The sub-triangular grid was used because it allowed us to write the closure condition of the linear system in an expedient fashion. Once the piecewise expression for $\mathbf{V}^*$ is known, it can be projected on the edges of the quadrilaterals and the sub-quadrilateral tracing can be employed. It is possible, however, to perform the tracing directly using the piecewise constant velocity field. This solution requires us to store the topology of a considerably larger triangular grid with five times as many elements as the quadrilateral grid.



(a) 4×4 Cartesian grid (b) Equivalent triangular grid

Figure 3.18: Homogeneous quarter of five spot: simulation grids

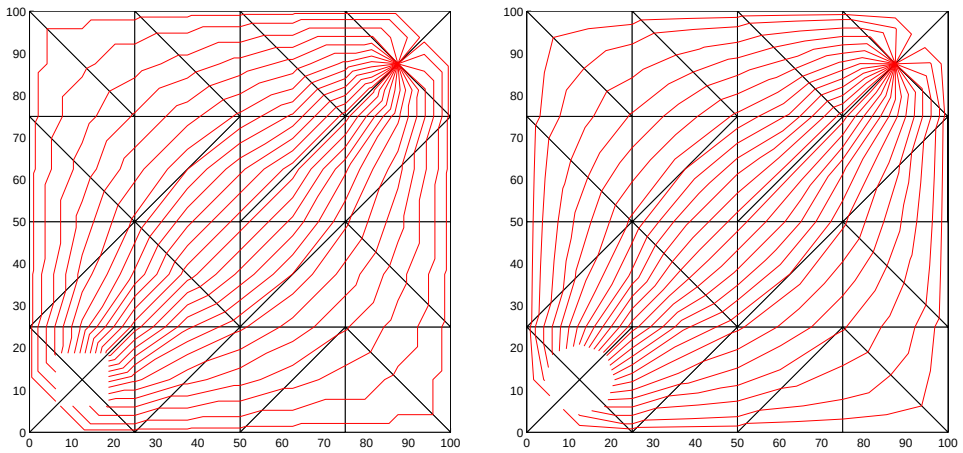
3.2.6 Summary of tracing results

Comparison tests are presented for the implemented tracing methods. The tracing and production curves for the tracer flow model are obtained using a CVFE pressure discretization and compared to the well known Pollock tracing for equivalent Cartesian models. The two tests considered are one homogeneous and one heterogeneous model of a quarter of a five spot.

Homogeneous quarter of a five spot

First, a quarter of a five spot is considered with a 4×4 Cartesian grid. The equivalent triangular grid is constructed by splitting every Cartesian grid cell (square) into two triangles. Cartesian grid well blocks were split into four in order to reproduce the same well locations (see Figures 3.18(a) and (b)).

Streamlines obtained with bilinear interpolation of velocity with continuous post-processed fluxes are shown in Figure 3.19(b) while Figure 3.19(a) shows the tracing with piecewise



(a) Tracing on a sub-triangular grid (b) tracing on a sub-quadrilateral grid

Figure 3.19: Cordes and Kinzelbach postprocessing, comparison of tracing types (from Prévost, 2000)

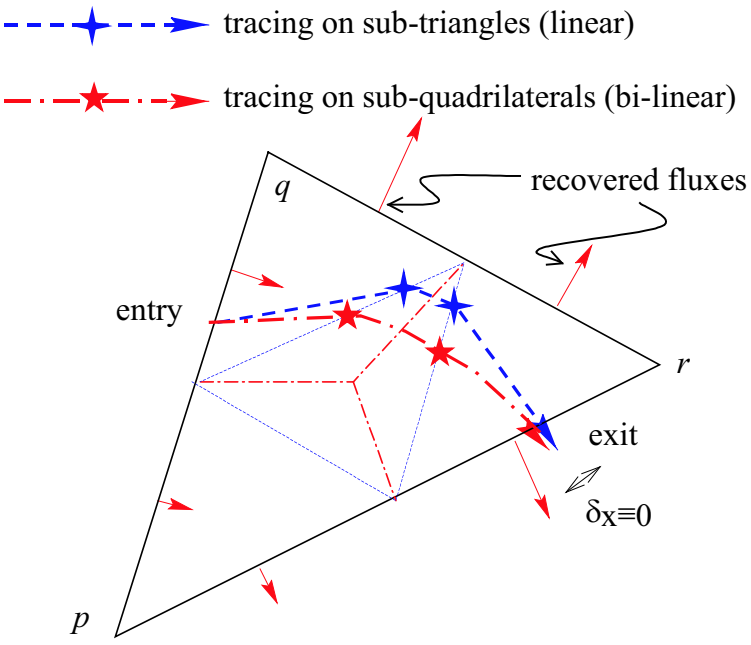
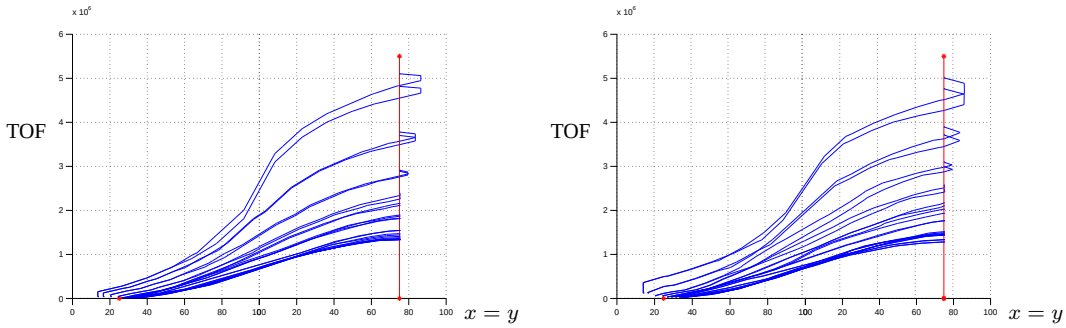


Figure 3.20: Entry and exit points using the post-processed fluxes are identical when using a piecewise constant (sub-triangle) or linear (sub-quadrilateral) approximation for the velocity



(a) Sub-triangular tracing (b) Sub-quadrilateral tracing

Figure 3.21: Time of flight versus longitudinal distance to injector (Cordes and Kinzelbach postprocessing applied to different tracing type)

constant velocity with the same post-processed fluxes. The two tracings are equivalent in terms of entry and exit points as tests indicate differences in exit and entry points coordinates of the order of the precision of the machine floating point representation ($\epsilon = 10^{-9}$). However, along each streamline, some small differences in time of flight (order 1%) can be observed between the two tracing methods as shown in Figures 3.21(a) and (b). The graphs were obtained by plotting for each streamline the curvilinear time of flight at a point as a function of the point location, projected on the $x = y$ axis (see also Figure 3.22).

The tracer flow front is plotted for different pore volume injected (PVI) in Figure 3.23 and shows a breakthrough time of 0.7 after accounting for well spacing differences (on a Cartesian grid, the well spacing is $\sqrt{2}(L - \Delta x)$ instead of $\sqrt{2}L$, where L is the side length of the domain).

When running the same problem with a finer underlying Cartesian grid (36×36), the time of flight proved to be in excellent agreement with simulations on Cartesian grids and the fractional flow curve matched the analytical solution with a dimensionless breakthrough time of 0.72.

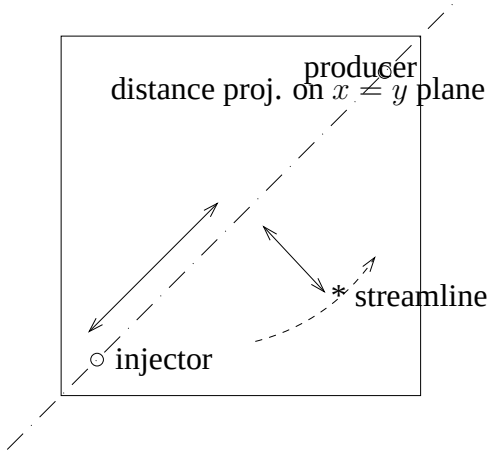


Figure 3.22: Time of flight on Figures 3.21(a) and 3.21(b) is a projection on the $x = y$ plane of time along the streamlines viewed as an elevation above the (x, y) plane

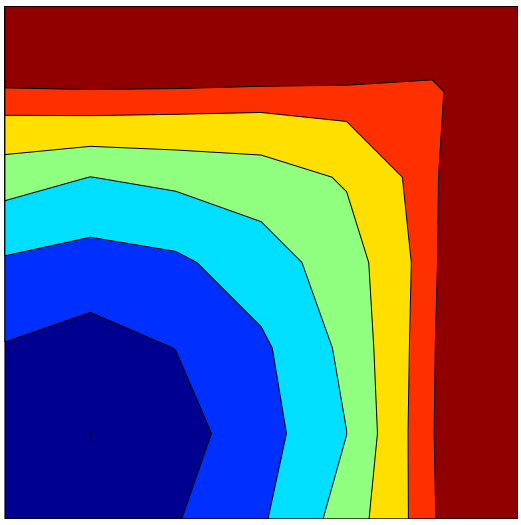


Figure 3.23: Tracer flow fronts at different pore volumes injected, $t_D = 0.15$, 0.33, 0.40, 0.53 and 0.66 (6×6 grid as shown in Figure 3.8)

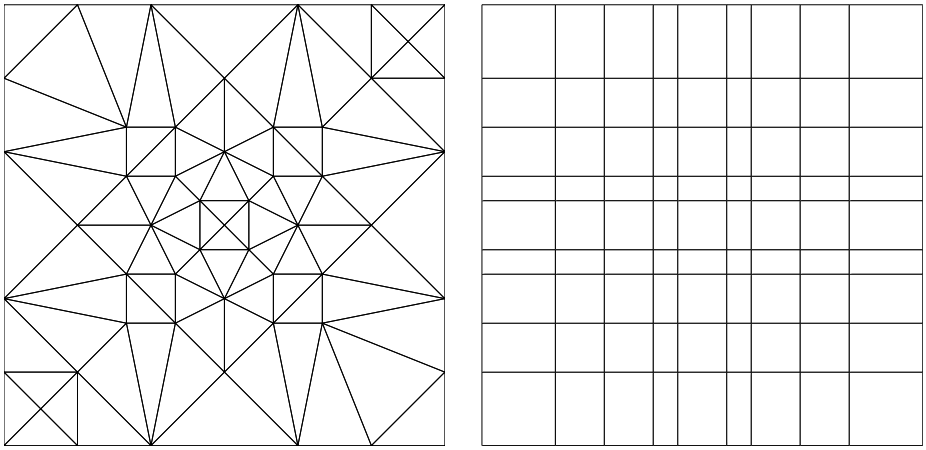


Figure 3.24: CVFE grid with cell based permeability is constructed to compare with an equivalent sized Cartesian grid

Heterogeneous quarter of a five spot

In order to compare the conventional streamline method on a Cartesian grid with the cell centered and point distributed permeability CVFE schemes on triangles for heterogeneous problems, a test case is carefully constructed such that the problem remains invariant with respect to each grid type.

The field consists of a uniformly high homogeneous isotropic permeability, with three square regions inserted with low permeability. The ratio between the high and low permeability regions is 10^3. The corresponding grids and control volumes are shown in Figures 3.24 and 3.25.

Cell centered approximation The pressure equation is solved using the standard CVFE discretization. After applying the flux postprocessing streamlines are traced using both piecewise constant (sub-triangular grid) and bilinear (sub-quadrilateral grid) velocity interpolations.

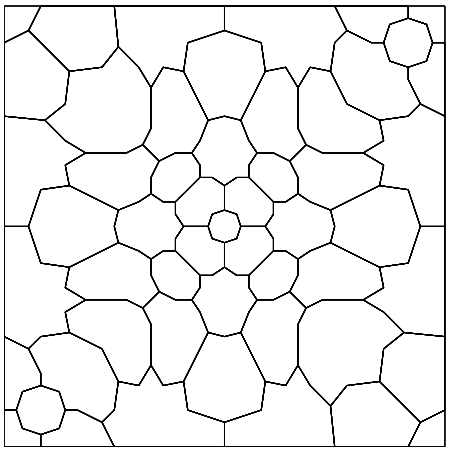


Figure 3.25: Dual CVFE grid. With cell based permeabilities, each control volume may contain different permeability values

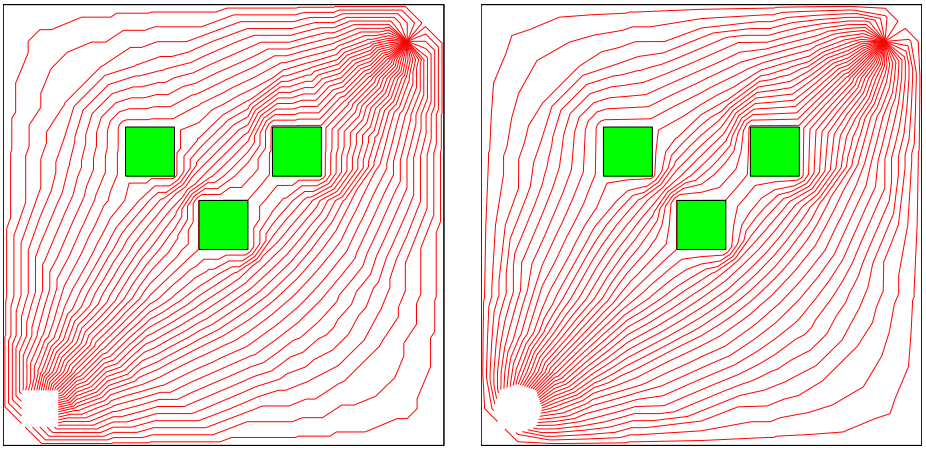


Figure 3.26: Streamline tracing for the cell distributed permeability case with flux postprocessing using a piecewise constant (sub-triangular grid on left image) or a bilinear (sub-quadrilateral grid on right image) velocity interpolation

Point distributed approximation To construct a point distributed grid that matches the same permeability discontinuities as in the previous section, the primal grid (shown in Figure 3.27(a)) is first introduced. Initially, the corresponding control volume grid (where permeability is defined; see Figure 3.27(c)) is not aligned with the interior permeability boundaries and mid-edge points are shifted (Figure 3.27(d)) to create a boundary aligned grid with the correct distribution. Both sub-triangle tracing and the new flux continuous sub-quadrilateral tracing are compared with Cartesian results.

Figure 3.28 shows that the proposed method respects the no flow boundaries and exhibits the same overall streamline distribution across heterogeneities as the Cartesian model. Plotting the fraction of streamlines breaking through against the dimensionless streamline arrival time gives the semi-analytical tracer fractional flow at the producing well (Figure 3.29). A non trivial difference in the breakthrough time is observed. This difference is attributed to the higher degree of refinement of the Cartesian grid (three cells between heterogeneities, see Figure 3.27(b)), compared to the control volume grid (one control volume cell only, see Figure 3.27(d)), which clearly favors the Cartesian solution.

3.2.7 Conclusions for 2D tracing

- For homogeneous and heterogeneous standard CVFE grids, in which permeabilities are defined over the triangles, the results obtained by Cordes and Kinzelbach (1992) using a sub-triangular grid were reproduced.

- Prévost (2000) noted that Cordes and Kinzelbach (1992) tracing could not be directly applied to the point distributed scheme, in which permeabilities are defined over control volumes. To circumvent that limitation, tracing on a sub-quadrilateral grid was proposed.

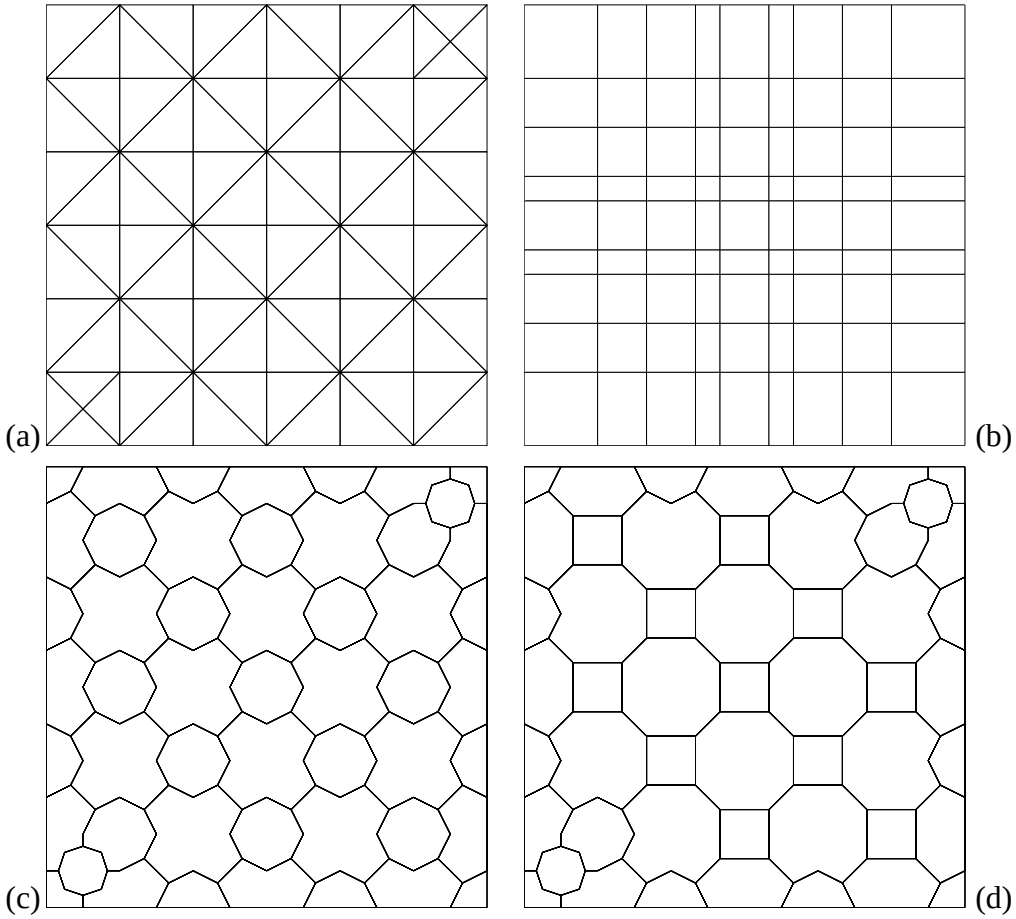


Figure 3.27: (a) Primal (triangular) grid and (c) corresponding CVFE grid. To match Cartesian control volumes and reproduce the same permeability discontinuities as in (b), the CVFE grid mid-edge points were shifted to give the grid in (d)

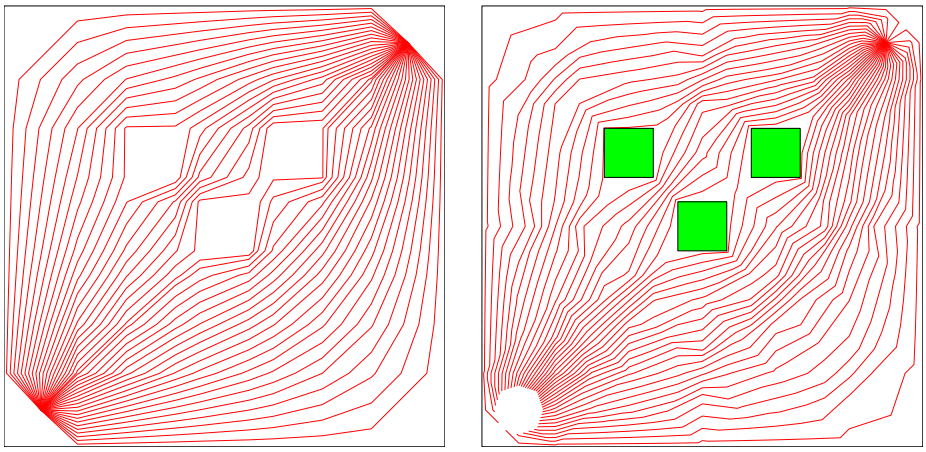


Figure 3.28: Tracing obtained from Cartesian simulation (left) is correctly reproduced by tracing on the sub-quadrilateral grid after flux post-processing (right)

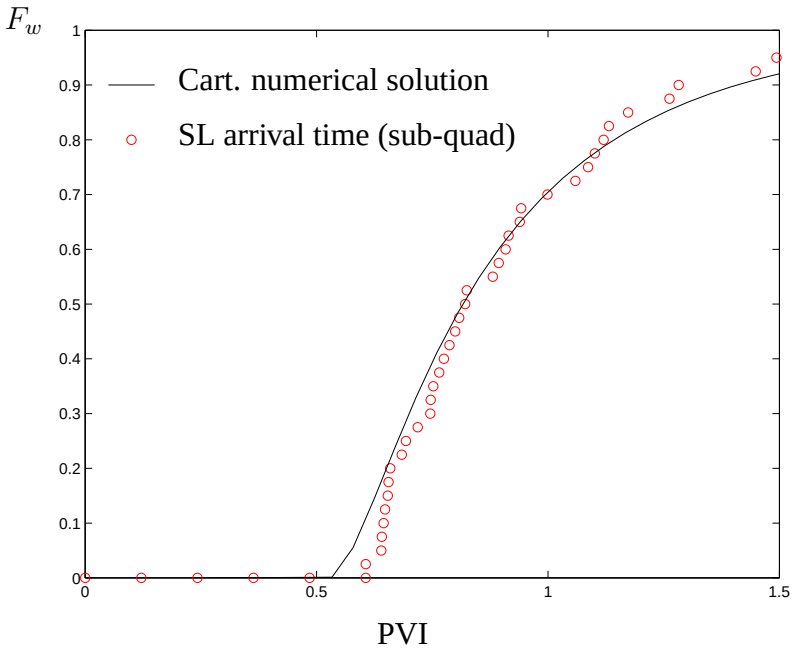


Figure 3.29: Tracer fractional flow. Cartesian numerical solution is compared to that obtained by plotting streamlines arrival times for the sub-quadrilateral tracing

- Tracing on a sub-quadrilateral grid (rather than using a sub-triangular grid) after performing a flux post-processing allows us to use the generalization of Pollock's tracing for non-orthogonal cells. Entry and exit points on primal triangular grid edges were unchanged and small differences were observed in terms of time of flight along the streamlines. This tracing technique is satisfactory for both the point-distributed and the cell-centered permeability approximations.

- An extension of Cordes and Kinzelbach (1992) sub-triangular tracing was proposed here for use with the point distributed scheme. The technique assumes a sub-division of the primal grid into six triangles, instead of four in the cell-centered approximation.

- For both the point-distributed and the cell-centered approximations, there exists both sub-triangular and sub-quadrilateral grid tracing methods that provide accurate tracing.

- Sub-triangular and sub-quadrilateral tracings are equivalent. Yet, the constant Jacobian approximation in the mapping of quadrilaterals to unit squares introduces a difference in the producing fractional flow curves.

3.3 Three dimensional tracing

This section generalizes the streamline technique to 3D. The method proposed introduces a sub-grid for the tracing and a local postprocessing that solves for the velocity interpolant. The postprocessing entails the resolution of an underdetermined linear system. System construction, size, rank, and closure techniques are each described in the present section.

3.3.1 Notation

The following notations are introduced to describe the post-processing:

CVFE Control volume finite element. Numerical method which consists in

- considering fluxes on the faces of a volume,
- expressing these fluxes as linear combination of unknown pressures,
- solving for these pressures by enforcing mass conservation within the volume.

Control volume Volume surrounding a pressure. Darcy fluxes are written at control volume boundaries.

Node Vertex of the primal grid. There is a one to one correspondance between nodes and control volumes, hence they are both indexed by the same letter p.

Hexahedron The intersection between the control volume p and the tetrahedron connected to the node p.

Tetrahedron refers to an element of the primal simulation grid. The pressure variables are located at the vertices of these elements.

Sub-division Each hexahedron is sub-divided into 5 tetrahedra (called sub-tetrahedra).

Sub-patch Collection of all the sub-tetrahedra belonging to both a given control volume and a tetrahedron (5 of them).

Patch Collection of all the tetrahedra lying in a control volume ($5n_t^p$ of them).

Sub-tetrahedron Refers to an element of a patch.

The following table describes the notations used in the present section.

Notation	Definition
Roman	
n_t^p	Number of tetrahedra connected to the node p (also see n_t)
n_t	Number of tetrahedra connected to a node
n_f	Number of triangular faces connected to a node
$\mathcal{H}$	Hexahedron
$\mathcal{T}$	Tetrahedron
$\mathcal{F}$	Triangular face
$\mathcal{Q}$	Quadrilateral face (face of a control volume)
$\mathcal{E}$	Edge
$\mathcal{M}$	Midpoint of edge
$\mathcal{C}$	Barycenter of a face
$\mathcal{G}$	Barycenter of a tetraheron
p	Vertex of the primal grid
$\hat{\mathbf{n}}$	Normal vector
Superscript	
p	Relative to a patch or equivalently to a node or a control volume p
t	Relative to a tetrahedron
f	Relative to a face
Subscript	

continued on next page

Notation	Definition
m	Index of a sub-tetrahedron within a patch
i, k	Interior sub-tetrahedron number $k = 1, 2, 3$ (always relative to a patch and a tetrahedron)
e	Exterior sub-tetrahedron (always relative to a patch and a tetrahedron)
c	Center sub-tetrahedron (always relative to a patch and a tetrahedron)
x, y, z	Physical space coordinates
ξ, η, ϵ	Reference space coordinates

3.3.2 Patch hexahedron sub-division

Each vertex p of the primal grid (Figure 3.30) defines a control volume of the dual grid. This control volume is comprised by the union of n_t hexahedra. Each hexahedron is defined by the eight following vertices (see also Figure 3.31):

- the vertex p,
- the centroid of the tetrahedron $\mathcal{G}$,
- the three centers of faces connected to p: $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$,
- the three centers of edges connected to p: $\mathcal{M}_{13}$, $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$.

Fundamental relation

When looking for a representation of the velocity in 2D, we saw that the number of degrees of freedom was proportional to the number of triangles connected to p, while the number

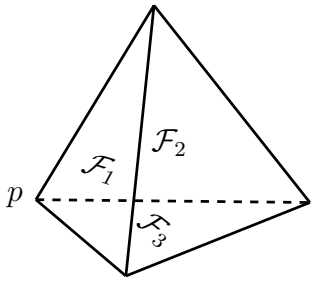


Figure 3.30: Definition of the tetrahedron faces (relative to p)

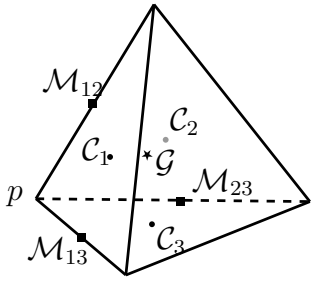


Figure 3.31: Definition of the points on the control volume (relative to p)

of constraints was proportional to the number of connected edges n_e. We took advantage of the fact that $n_f = n_e$ to obtain a full-rank system.

In 3D, away from the boundaries, the following relation exists between the number of faces and the number of tetrahedra connected to node p:

$$3n_t = 2n_f \tag{3.16}$$

The number of degrees of freedom of our expression for the velocity will be proportional to n_t, while the number of constraints will be proportional to n_f.

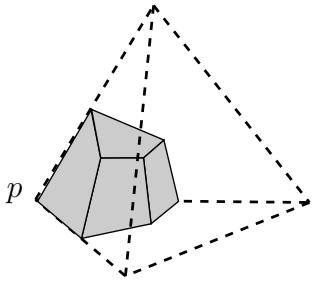


Figure 3.32: Sub-hexahedron $\mathcal{H}$

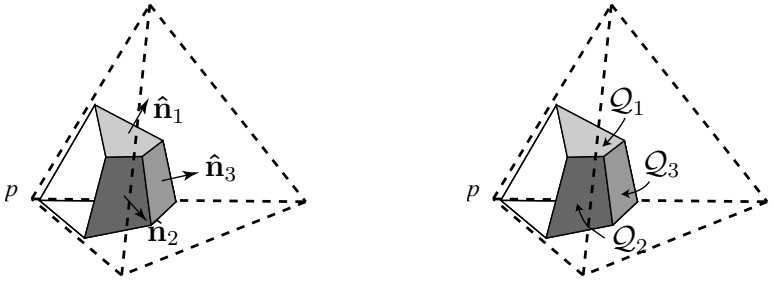


Figure 3.33: Fluxes on the control volume faces define the natural velocity

3.3.3 System of constraints

We can construct a system of equations similar to the one obtained for the 2D case. The velocity is now assumed to be a trilinear function defined on the unit cube. Velocity for a hexahedron in the physical coordinate system is again obtained via an isoparametric transformation. There are six degrees of freedom (dof) associated with the velocity expression, they are the values of the flux on each face of the hexahedron. The velocities on the n_t hexahedra forming the patch centered around the node p are constrained using:

- the $3n_t$ external fluxes (consistency constraints),
- the n_f internal continuity constraints,
- n_t constraints expressing that the node is neither a source or a sink.

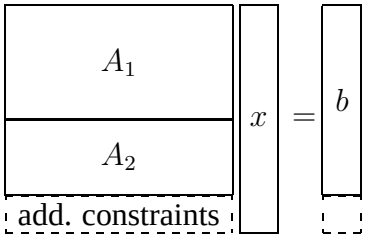


Figure 3.34: System of constraints for the trilinear interpolation on hexahedra. A_1 and A_2 represent respectively the continuity and zero-sum constraints of the fluxes

The consistency constraints are identified to the corresponding dof's in a straightforward manner. From the balancing of the dof's and the constraints we see that there remains a total of $3n_t$ dof's and $n_t + n_f$ constraints. Using Equation 3.16 we see that there are $n_f - n_t = n_t/2$ too many degrees of freedom (Figure 3.34).

To close the system of equations we can consider different options, some of which will be discussed in Section 3.3.4. One option is to enforce that the velocity derives from a potential in a weak sense by integrating $(\nabla \times \mathbf{V}^*) \cdot \hat{\mathbf{n}}$ on a series of surfaces. Again, for the reasons stated earlier, it is convenient to work with an analytical expression of $\mathbf{V}^*$, hence we choose to subdivide the patch further in order to use piecewise constant approximations for $\mathbf{V}^*$.

Patch sub-tetrahedra division

Each hexahedron belongs to both a control volume (p) and a tetrahedron $\mathcal{T}$. We divide each hexahedron in the following manner:

$\mathcal{T}_{\mathrm{e}}$:	the sub-tetrahedron connecting the center of T (*i.e.*, $\mathcal{G}$) to the centers of the three faces connected to p (*i.e.*, $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$)
$\mathcal{T}_{\mathrm{c}}$:	the sub-tetrahedron connecting p to the centers of the three faces connected to p
$\mathcal{T}_{\mathrm{i},j},\ j \in \{1,2,3\}$:	three sub-tetrahedra completing the hexahedra

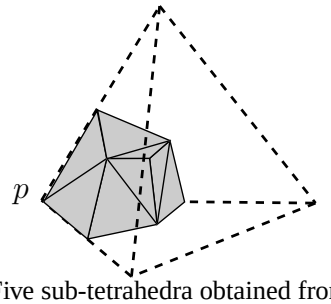
An illustration of the subdivision is provided in Figure 3.3.3.

Flux density

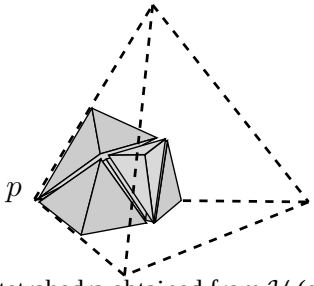
Each flux value on the face of the control volume acts as a consistency constraint for the velocity expression in the sub-elements touching the face. This constraint translates into a condition on the normal component of the velocity at the face. Assuming that the normal component of the velocity is uniform on the face simplifies considerably the writing of the system of constraints. Indeed, it decouples the system for each patch while still enforcing continuity of the normal component of the velocity at *all points* across all control volume faces.

Using this assumption, we can define a face flux-density, equal by definition to the value of the flux across the face divided by the area of the face. The consistency constraint now reads “the normal component of the velocity equals the face flux density at each point”.

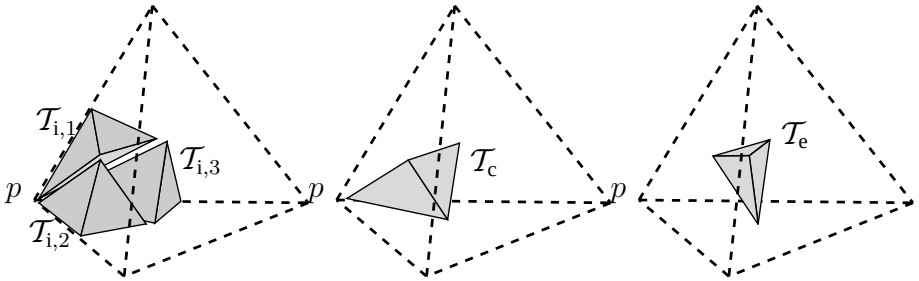
$$\mathbf{V}^* \cdot \hat{\mathbf{n}} = f^\circ / |\mathcal{Q}| \tag{3.17}$$



(a) Five sub-tetrahedra obtained from $\mathcal{H}$



(b) Five sub-tetrahedra obtained from $\mathcal{H}$ (exploded view)



(c) Five sub-tetrahedra obtained from $\mathcal{H}$

Figure 3.35: Subdivision of an hexahedron $\mathcal{H}$ into five subtetrahedra

Consistency constraints on the sub-tetrahedra

From Figures 3.35(a) and (c), we see that each face of the control volume was split into two triangles and and each of these triangles belongs to two distinct sub-tetrahedra ($\mathcal{T}_{\mathrm{e}}$ and $\mathcal{T}_{\mathrm{i},j}$). As anticipated, the flux on the control volume face $\mathcal{Q}_j$ acts as a consistency constraint on the value of the velocity in the two sub-tetrahedra: $\mathbf{V}^*_{\mathrm{e}}$ and $\mathbf{V}^*_{\mathrm{i},j}$. We need still to decide how the flux constraint on $\mathcal{Q}_j$ is distributed on the two triangles. A reasonable assumption is that the flux-density is the same on both triangles. Any other assumption might introduce a coupling between velocities of different patches hence defeating the purpose of a local post-processing.

Use of the natural velocity

The natural velocity is defined in the same way as in 2D; specifically, as the vector that reproduces the numerically computed fluxes across the control volume faces. The natural velocity is denoted $\mathbf{V}^\circ$ and is reconstructed from f°_j, ($j = 1:3$) using:

$$\begin{cases} \mathbf{V}^\circ \cdot \hat{\mathbf{n}}_1 &= f^\circ_1 / |\mathcal{Q}_1| \\ \mathbf{V}^\circ \cdot \hat{\mathbf{n}}_2 &= f^\circ_2 / |\mathcal{Q}_2| \\ \mathbf{V}^\circ \cdot \hat{\mathbf{n}}_3 &= f^\circ_3 / |\mathcal{Q}_3| \end{cases} \tag{3.18}$$

(see Figure 3.33 for notation).

The consistency constraints applied to the exterior sub-tetrahedron $\mathcal{T}_{\mathrm{e}}$ immediately give

$$\mathbf{V}^*_{\mathrm{e}} = \mathbf{V}^\circ \tag{3.19}$$

In addition to that, the velocities in the three interior sub-tetrahedra are constrained:

$$\mathbf{V}^{*}_{\mathrm{i},j} \cdot \hat{\mathbf{n}}_j = f^{\circ}_j / \left|\mathcal{Q}_j\right| \quad \text{with} \quad j = 1:3 \tag{3.20}$$

Continuity constraints

A sub-tetrahedral patch is composed of n_t sub-patches. Each sub-patch is composed of five sub-tetrahedra. The velocity in the sub-tetrahedron $\mathcal{T}_{\mathrm{e}}$ is already determined (see Equation 3.19). The velocities in the three interior sub-tetrahedra ($\mathcal{T}_{\mathrm{i},1}$, $\mathcal{T}_{\mathrm{i},2}$, and $\mathcal{T}_{\mathrm{i},3}$) are constrained by one consistency equation each, corresponding to the face that lies on the control volume envelope. Finally, the central sub-tetrahedron $\mathcal{T}_{\mathrm{c}}$ does not share any face with the control volume envelope, and hence there are no consistency constraints on the central sub-tetrahedron velocity.

Continuity of the normal component of the velocity must be enforced across each of the sub-tetrahedra faces within the patch. For each sub-patch, there are

- 6 sub-tetrahedra faces on the outside of the sub-patch ($\mathcal{T}_{\mathrm{i},j}$)
- 3 sub-tetrahedra faces connecting the $\mathcal{T}_{\mathrm{i},j}$ to $\mathcal{T}_{\mathrm{c}}$
- 1 sub-tetrahedron face connecting the $\mathcal{T}_{\mathrm{c}}$ to $\mathcal{T}_{\mathrm{e}}$

Hence in total for the patch, there are $2n_f + 4n_t$ continuity constraints.

We summarize the number of unknowns and constraints for a patch:

Constraints		**Unknowns**
consistency	$3n_t$	$4 \cdot 3 \cdot n_t$
continuity	$2n_f + 4n_t$	$= 12n_t$

For a node in the interior of the gridded domain, using the relation between n_t and n_f (Equation 3.16), the size of the system is $10n_t \times 12n_t$.

System of constraints

In order to write a matrix form of the system of constraints, we define the following unknown compound-vector, comprised of all the velocities defined in the sub-tetrahedra of the patch:

$$x = \left[x^1, \cdots, x^{n_t}\right]^T \tag{3.21}$$

with

$$x^k = \left[x_1^k, x_2^k, x_3^k, x_4^k\right]^T \quad k \in \{1, \cdots, n_t\} \tag{3.22}$$

and

$$\begin{cases} x_1^k &= \mathbf{V}_{\mathrm{i},1}^{*,k} \\ x_2^k &= \mathbf{V}_{\mathrm{i},2}^{*,k} \\ x_3^k &= \mathbf{V}_{\mathrm{i},3}^{*,k} \\ x_4^k &= \mathbf{V}_{\mathrm{c}}^{*,k} \end{cases} \tag{3.23}$$

The system of constraints is then written as $Ax = b$ (Figure 3.36). A is $10n_t \times 12n_t$ and can separated into an upper part of size $3n_t \times 12n_t$ representing the consistency constraints and a lower part of size $7n_t \times 12n_t$ representing the continuity constraints. The upper part of b is composed of the right hand side of the consistency constraints (Equation 3.20) and the lower part is zero as the continuity constraints do not introduce a right hand side.

The rank of A is necessarily less than $10n_t$. Because one consistency constraint is redundant, the rank is at most $10n_t - 1$. As a consequence, the rank deficiency of A (equal to the dimension of its null space) is equal to at least $2n_t + 1$. Numerical tests showed that the rank is (in general) exactly equal to $10n_t - 1$. To solve the system, one possibility

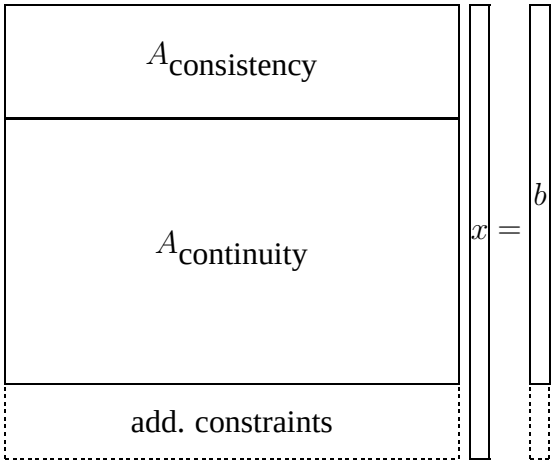


Figure 3.36: System of constraints for the piecewise constant interpolation on sub-tetrahedra. Additional constraints may be added to close the system

is to remove one consistency constraint from the system and impose $2n_t + 1$ additional constraints in order to complete the system and obtain a full rank matrix. Alternatively, the system can be solved in a least square sense. These alternatives are discussed in the following section.

3.3.4 Remarks

System size, square system

If we also assume that the velocity in the sub-tetrahedron $\mathcal{T}_c$ is known and equal to the natural velocity ($\mathbf{V}_c^* = \mathbf{V}^\circ$), then the system of constraints becomes:

Constraints		**Unknowns**
consistency	$3n_t$	$3 \cdot 3 \cdot n_t$
continuity	$2n_f + 3n_t$	$= 9n_t$

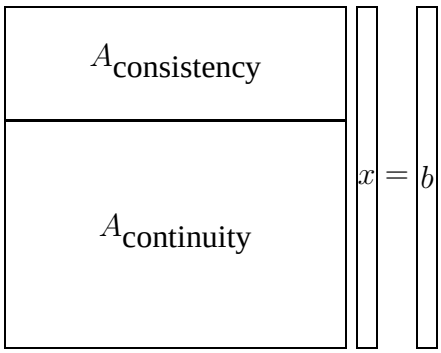


Figure 3.37: System of constraints for the piecewise constant interpolation on sub-tetrahedra. Although the system appears to be closed, it is actually incompatible

For a node in the interior of the gridded domain, using the relation between n_t and n_f (Equation 3.16), the size of the system becomes $9n_t \times 9n_t$. Its rank, however, can vary and is *not* in general $9n_t - 1$.

Moreover, numerical tests showed that the system right-hand side (containing the flux density values) does not usually lie in the image of the matrix, leading to an incompatible system (no solution). The shape of the system is summarized in Figure 3.37. The assumption $\mathbf{V}_{\text{c}}^{*} = \mathbf{V}^{\circ}$ is therefore not viable as it results in a linear system that generally does not have a solution.

Closure of the system

For an interior point, the remaining $2n_t + 1$ degrees of freedom are used to further constrain the solution. Consistent and continuous velocities may produce nonphysical results due to

nonzero curl of the velocity, which is not automatically enforced. To illustrate the importance of zero curl velocities, we now set the value of the curl of the velocity to an arbitrary nonzero value. In Figure 3.38, the curl of the velocity was forced to be large (of the order of the velocity itself) in certain cells. This was achieved using Stoke's theorem (given below) and writing the circulation of the velocity on a closed line on the surface of the patch, resulting in a linear constraint on the velocity components. The problem simulated is a homogeneous, corner to corner flow and should therefore display smooth streamlines. However, the corresponding streamline tracings exhibit swirling and nonphysical delays in the time of flight. This illustrates the potential benefit of controlling the magnitude of the curl of the postprocessed velocities. We now describe how this can be achieved.

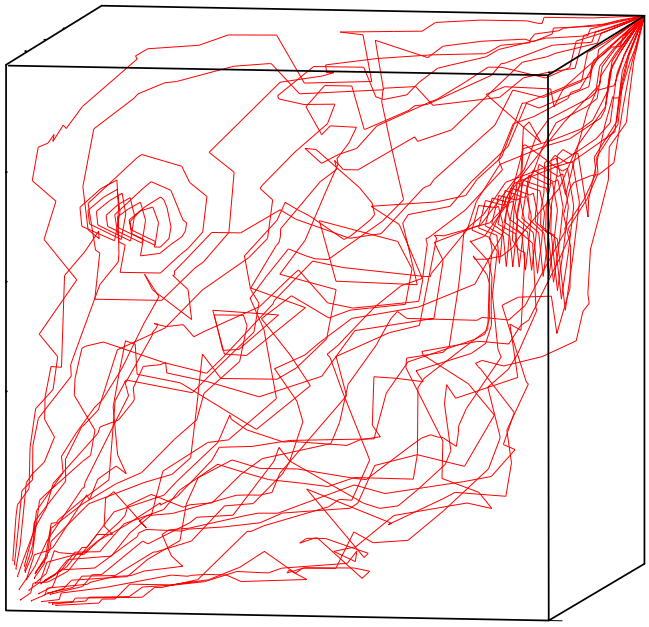


Figure 3.38: Consistency and continuity constraints alone may produce nonphysical swirling of the streamlines (here $\nabla \times \mathbf{u}$ is artificially forced to be nonzero)

In the case where the underdetermined system of constraints 3.36 is solved in a least

square sense (no additional constraints added), numerical tests showed that the circulation of the velocity interpolant along any given closed curve was in geneneral nonzero. As a consequence, and in order to prevent the possible nonphysical swirling of the streamlines (comparable in nature to that exhibited in Figure 3.38 but generally much smaller in magnitude), additional zero-curl constraints are added to the system.

Because $\mathbf{V}^*$ is not differentiable everywhere, we only ensure it is weakly non-rotational in the three coordinate directions. Using Stoke's theorem

$$\iint \nabla \times \mathbf{u} \cdot \mathbf{d}\hat{\mathbf{n}} = \oint \mathbf{u} \cdot \mathbf{d}\hat{\mathbf{t}} \tag{3.24}$$

for any vector $\mathbf{u}$, where $\mathbf{d}\hat{\mathbf{t}}$ is a elementary vector tangential to the oriented curve, we can calculate the circulation of $\mathbf{V}^*$ on a set of closed curves on the surface of the patch. This analytical expression is a linear function of the components of the velocities of the sub-tetrahedra intersected by the curve.

Let us consider the three planes $\mathcal{P}_u$, ($u \in \{x, y, z\}$), defined as the planes of constant coordinate u containing the center p of the patch (see Figure 3.39). The intersections of these planes with the control volume envelope define three curves that we use to enforce the zero-curl constraints.

The remaining $2n_t - 2$ degrees of freedom can be treated in different ways:

- Force the expression of the circulation of the velocity to be zero for additional curves (Figure 3.40).

- Minimize the sum of the norms of the velocities.

- Minimize the deviation of $\mathbf{V}^*$ from: $\mathbf{V}^\circ$

$$\sum_{m=1}^{n_t}\left[\sum_{j=1}^{3}\|\mathbf{V}_{i,j}^{*,m}-\mathbf{V}^{\circ,m}\|^2+\|\mathbf{V}_{c}^{*,m}-\mathbf{V}^{\circ,m}\|^2\right] \tag{3.25}$$

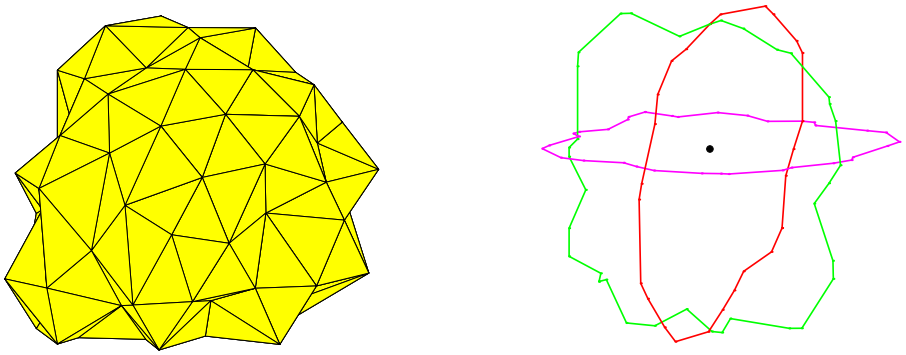


Figure 3.39: Closure conditions can be obtained by setting the circulation of the velocity on some path on the surface of the patch to zero

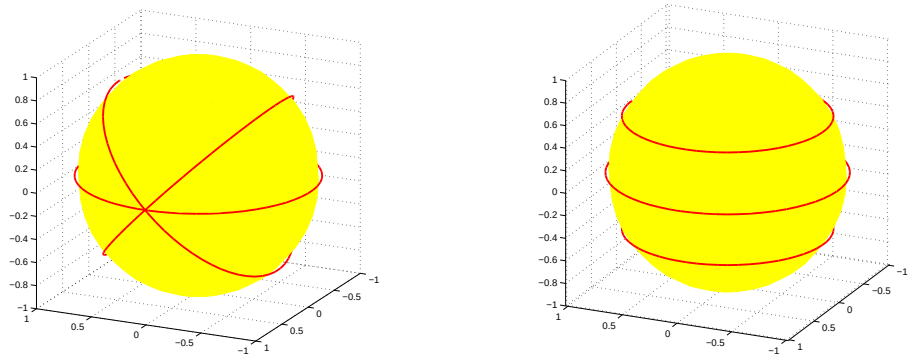


Figure 3.40: In addition to the three paths obtained by intersecting the patch with the x, y and z planes, other paths may also be considered

Including the three irrotational constraints plus the norm-minimization objective constraint leads to the system shape shown in Figure 3.41. The resulting tracings are satisfactory and displacement data agree with the reference solution for tracer flow on a quarter of

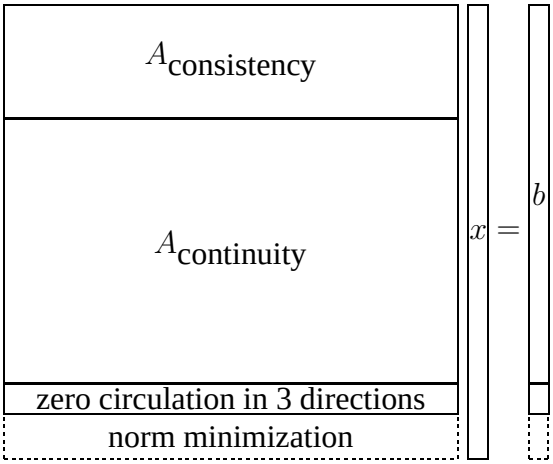


Figure 3.41: System of constraints for the piecewise constant interpolation on subtetrahedra. Irrotational constraints are added to close the system and the remaining degrees of freedom are used to minimize the deviation between $\mathbf{V}^*$ and $\mathbf{V}^\circ$

a nine spot (Figure 3.42). Detailed results for the streamline tracings will be presented in Chapter 5.

Solution of the system

If the system of linear constraints is underdetermined or nearly singular, special care must be taken in the inversion procedure. As explained in the current section, a full rank system of constraints is constructed in order to impose physically meaningful constraints on the velocity representation. However, in some cases, redundant or even incompatible constraints may be imposed on the system, leading to rank deficient matrices. As a consequence, solution methods that are more sophisticated than standard Gauss inversion or iterative methods are needed. Common methods for inverting underdetermined systems include orthogonal factorization and singular value decomposition methods. The use of these methods introduces an extra cost to the inversion, but provides critical control on the norm of the solution

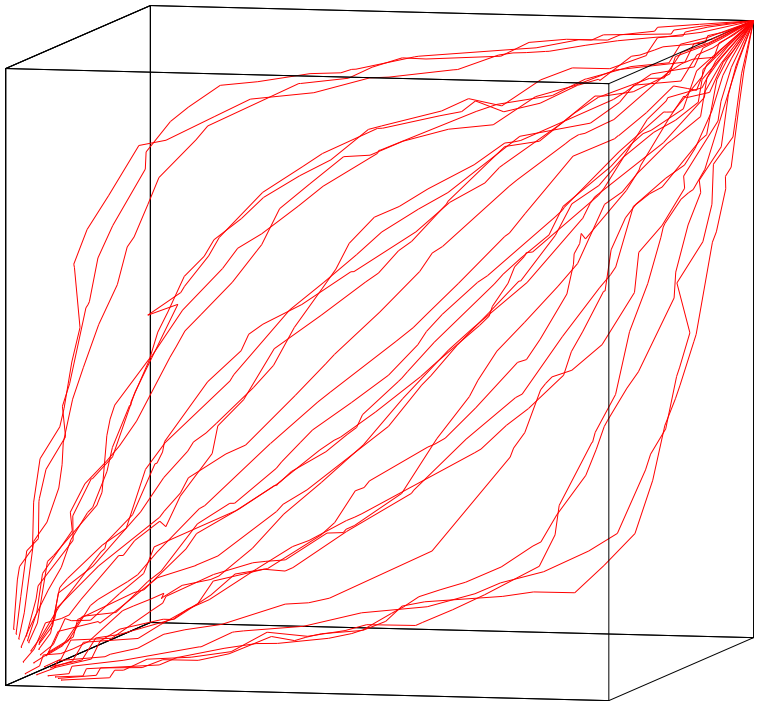


Figure 3.42: Enforcing modified system of constraints gives non-swirling streamlines

and permits the generation of meaningful results for nearly singular systems. We describe here two methods that can be used to solve an underdetermined system of constraints in a least square sense, the QR factorization method and the singular value decomposition (SVD) method. Both methods provide a solution that minimizes the 2-norm among possible solutions of the underdetermined system. Comparative benefits of the two methods are discussed as well as an assessment of computational costs.

QR factorization The QR factorization of an $m \times n$ matrix is given by

$$A = QR \tag{3.26}$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal ($Q^TQ = QQ^T = I$) and $R \in \mathbb{R}^{m\times n}$ is upper triangular (here we assume $m \geq n$). If A has full column rank, then the first n columns of Q form an orthogonal basis for the range of A. The computation hence amounts to determining an orthonormal basis for a set of vectors. Several techniques can be used for the factorization; e.g., Householder factorization or Gram-Schmidt factorization. If A is underdetermined ($m < n$), then some factorization algorithm can provide the minimum 2-norm solution. If we compute the QR factorization,

$$A^T = QR = R \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \tag{3.27}$$

with $R_1 \in \mathbb{R}^{m\times m}$, then $Ax = b$ becomes

$$(QR)^T x = \begin{bmatrix} R_1^T & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = b \tag{3.28}$$

where

$$Q^T x = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad z_1 \in \mathbb{R}^m, z_2 \in \mathbb{R}^{n-m} \tag{3.29}$$

z_1 is therefore uniquely defined and z_2 can take an arbitrary value. Setting $z_2 = 0$ will then minimize the 2-norm of x because

$$\|Q^T x\|_2{}^2 = \|x\|_2{}^2 = \|z_1\|_2{}^2 + \|z_2\|_2{}^2$$

The least square solution is then

$$x_{LS} = R_1^{T^{-1}} b \tag{3.30}$$

Singular Value Decomposition (SVD) The SVD can also be used to compute the minimal norm solution of an underdetermined $Ax = b$ problem. For a real $m \times n$ matrix, there exist orthogonal matrices,

$$U = [u_1, \cdots, u_m] \in \mathbb{R}^{m \times m} \quad \text{and} \quad V = [v_1, \cdots, v_n] \in \mathbb{R}^{n \times n} \tag{3.31}$$

such that

$$U^T A V = \text{diag}(\sigma_1, \cdots, \sigma_p) \in \mathbb{R}^{m \times n} \quad p = \min\{m, n\} \tag{3.32}$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$.

The σ_i are the *singular values* of A and the vectors u_i and v_i are the ith *left singular vector* and the ith *right singular vector* respectively.

The SVD reveals important information about the structure of the matrix. In particular it gives the rank r of the matrix as the number of nonzero singular values,

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0 \tag{3.33}$$

Then,

$$\begin{aligned} \text{rank}(A) &= r \\ \text{null}(A) &= \text{span}\{v_{r+1}, \cdots, v_n\} \\ \text{range}(A) &= \text{span}\{u_1, \cdots, u_r\} \end{aligned} \tag{3.34}$$

and we have the SVD *expansion*

$$A = \sum_{i=0}^{r} \sigma_i u_i v_i^T \tag{3.35}$$

When dealing with rounding errors in the matrix, SVD is very useful in determining

whether the rank is numerically deficient by defining the ϵ-rank:

$$\text{rank}(A, \epsilon) = \min_{\|A-B\|_2 \leq \epsilon} \text{rank}(B) \tag{3.36}$$

An interesting property is that, if $r_\epsilon = \text{rank}(A, \epsilon)$, then

$$\sigma_1 \geq \cdots \geq \sigma_{r_\epsilon} > \epsilon \geq \sigma_{r_\epsilon+1} \geq \cdots \sigma_p \quad p = \min\{m, n\} \tag{3.37}$$

The ϵ-rank simply corresponds to the number of singular values that are greater than some tolerance ϵ.

Finally, the SVD provides a compact expression for the least square solution

$$x_{LS} = \sum_{i=1}^{r} \frac{u_i^T b}{\sigma_i} v_i \tag{3.38}$$

where r is typically an ϵ-rank. Note that ignoring small singular values amounts to setting $1/\sigma_i$ to zero, therefore improving the numerical stability of the solution.

Cost comparison Cost comparisons of the QR factorization and SVD are given below (Golub and Van Loan, 1996) for a square full system[1]

Method	**Flops**
Gauss Elimination	$2n^3/3$
QR (Householder Orthogonalization)	$4n^3/3$
Singular Value Decomposition	$12n^3$

[1] also assuming that the right-hand side is available at the time of factorization

SVD is about nine times more costly than QR factorization. Therefore, QR factorization is the preferred method for post-processing the velocity. Nevertheless, SVD provided very valuable information for the analysis of the problem and for the determination of the most appropriate patch subdivision.

Nodes on the boundary

For nodes on the boundary, Equation 3.16 does not hold and the number of additional degrees of freedom can drop below three. In that case, the solution space of the system can be found analytically (using SVD for instance) and the solution of the problem can be obtained by performing a minimization of the norm of the curl. For instance, if the number of additional degrees of freedom is equal to two, then

$$x = x_0 + \lambda_1 v_{n-1} + \lambda_2 v_n \tag{3.39}$$

where v_{n-1} and v_n are the $(n-1)$th and nth right singular vectors of A and λ_1 and λ_2 are arbitrary real numbers.

The algebraic expressions of the three components of the curl of the velocity on the intersecting curves (Figure 3.39) can be written as before. Because the rank deficiency is only two, the three linear expressions cannot all be forced to zero. The solution we consider here is the plane-minimization of the norm of the computed algebraic curl vector. An alternative solution is to enforce zero curl on only two curves.

Closure on the sub-hexahedra system

We saw from Figures 3.34 and 3.36 that the system size increases from $3n_t$ to $12n_t$ because three *a priori* unknown vector components are introduced in 4 sub-tetrahedra instead of

just 3 fluxes in the sub-hexahedral grid. This significant increase in computational expense could be avoided if we could directly write closure conditions as linear combinations of the fluxes in the sub-hexahedra. This requires the transformation of the expression of the circulation from the reference cube to physical space. This transformation is complicated because it requires the inversion of a nonlinear expression of the Jacobian of the transform. A substantial speed up of the velocity postprocessing will be obtained if this is achieved.

Chapter 4

3D Flow based upscaling and gridding

This chapter generalizes and extends to 3D the methods developed in Chapter 2 for the 2D case. Implementation of the 3D flow-based upscaling of both permeability tensors and two-point transmissibilities for unstructured grids is presented. As discussed below, the overall methodology is much more robust when two-point (as opposed to multiple-point) fluxes are used. Tracer-flow streamline simulation (as developed in Chapter 3) is then applied to provide flow diagnostics and to extract flow information for 3D models.

The significant increase in reservoir complexity when going from 2D to 3D models calls for more sophisticated gridding tools and algorithms (Mallet, 2002). These algorithms must provide grids that conform to complex geological features such as faults, pinchouts and seismic horizons (*e.g.*, Figure 4.1). Extra complexity also arises from the need to model nonconventional wells with varying trajectories and multilaterals.

This work makes use of gridding and visualizations tools developed by the LIAD[1] within the context of the Gocad consortium. In particular, recent advances of the Gocad group in conforming tetrahedrization has made new tools available. We have extended

[1]Laboratoire Infographie et Analyse des Données

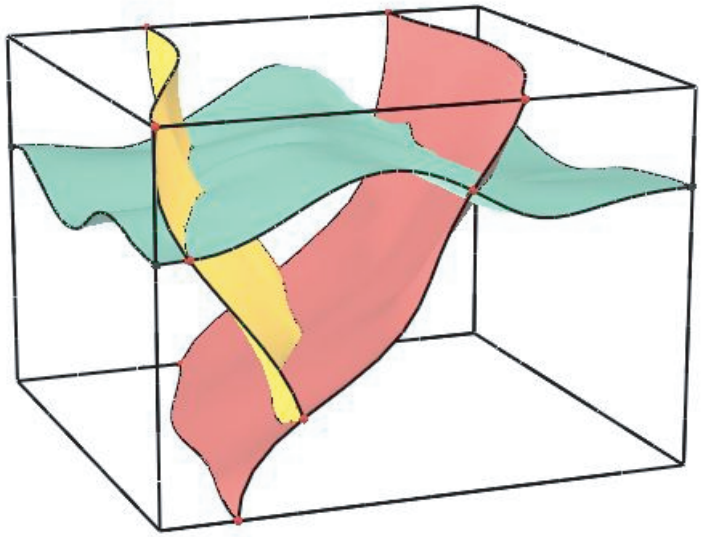


Figure 4.1: "Y" fault structural model (part of the Xingu structural model, courtesy of Gocad)

these tools to fit the specific requirements for the construction of 3D reservoir models suitable for reservoir simulation.

The first section of this chapter presents the extension of the flow based permeability tensor upscaling. Section 4.2 applies similar techniques to the upscaling of two-point transmissibilities. Section 4.3 describes the StanLab software development platform which is based on a Gocad kernel, as well as the different extensions developed by the LIAD that we use in this work, and a brief description of the gridding algorithms used in the TGridLab plug-in (one of the extensions to Gocad). Section 4.4 describes the grid construction and adaptation methodology. This entails the definition of a parameterized grid resolution constraint obtained from flow information maps.

4.1 Flow based upscaling

The permeability upscaling procedure described in Section 2.3.1 can be generalized to 3D unstructured grids, as we now describe. The upscaled permeability will in general be a full tensor quantity which will require a multipoint approximation. We assume for simplicity that the underlying geocellular model is defined on a uniform Cartesian grid.

4.1.1 Procedure

We now require three sets of flow problems to determine the equivalent permeability tensor. The upscaling procedure is as follows:
For each vertex p of the primal tetrahedral grid,

1. Consider the list $\left\{\mathcal{T}_j^p, j \in 1, \cdots, n_t^p\right\}$ of the tetrahedra connected to p

2. Identify the *target* region as the subset of the fine cells that are enclosed within the control volume centered on p

3. Construct the extended Cartesian region (target cell plus border region) $\mathcal{R}(I, J, K)$ where $I = [i_{\min}, i_{\max}]$, $J = [j_{\min}, j_{\max}]$ and $K = [k_{\min}, k_{\max}]$ are index intervals

4. Solve the single phase pressure equation on the extended region using appropriate boundary conditions. Three such solutions, with pressure drop specified in each of the coordinate directions are required

5. Compute $\mathbf{k}^*$ from the average velocities and average pressure gradients over the target volume

6. Enforce positive definiteness if necessary

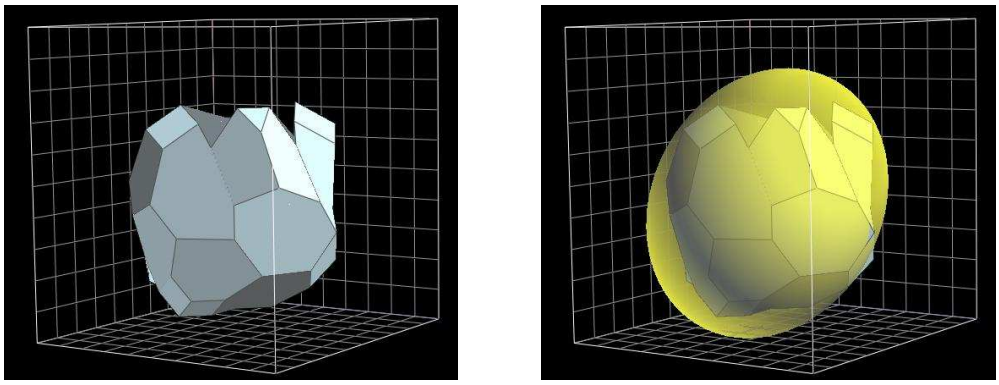
Target region

The target region is defined as the collection of fine grid cells that are used in the volume averaging for the determination of $\mathbf{k}^*$ (Equation 2.7). This region is a subset of the extended region over which the pressure equation is solved. A natural choice for the target region is the ensemble of all fine grid cells that are within the control volume considered. Because the exact geometry of the control volume can be very complex as shown in Figure 4.2(a), the procedure of identifying these cells (known as rasterization) can be very costly. This procedure is even more complex because of the fact that the control volume is in general non-convex.

An alternative to the exact determination of the target region is to approximate the control volume by an ellipsoid. Once an ellipsoid is fit to the control volume, all the fine Cartesian cell centers of the extended region can be transformed into the local principal coordinate axes of the ellipsoid. Testing whether or not the center points of the fine cells fall within the ellipsoid is straightforward. The determination of the ellipsoid entails finding its principal axes, which are not assumed to be parallel to the x, y or z axes.

Ellipsoid fitting

The technique used for the fitting of an ellipsoid around the control volume is based on principal component analysis (PCA). Considering a sampled spatial random variable, PCA finds the direction of maximum and minimum dispersion of the distribution in space (Johnson and Wichern, 1988). The principal axes of the ellipsoid are given by the eigenvalues of



(a) Non-convex dual polyhedral cell

(b) Ellipsoid fit to the geometry of the envelop

Figure 4.2: Approximation of the averaging volume for the calculation of $\mathbf{k}^*$

the covariance matrix:

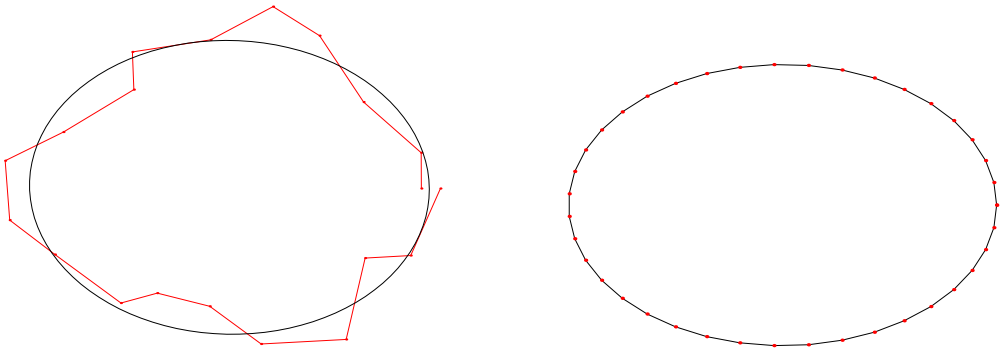
$$R = \begin{bmatrix} \sigma^2_{x,x} & \sigma^2_{x,y} & \sigma^2_{x,z} \\ \sigma^2_{y,x} & \sigma^2_{y,y} & \sigma^2_{y,z} \\ \sigma^2_{z,x} & \sigma^2_{z,y} & \sigma^2_{z,z} \end{bmatrix} \tag{4.1}$$

where

$$\sigma^2_{u,v} = \frac{1}{N}\sum_i (u_i - \langle u \rangle)(v_i - \langle v \rangle), \quad (u,v) \in \{x,y,z\}^2 \tag{4.2}$$

Assuming that the variable follows a trivariate Gaussian distribution, the lengths of the three axes of the ellipsoid are defined as 1.96 times the standard deviation in the respective directions, so as to include 95% of the sampled data along each axis.

In order to fit the control volume surface, we consider all the points defining the surface and perform a PCA. Clearly, the assumption of a Gaussian distribution does not apply to the distribution of the control volume points. Therefore, the center of the control volume is added to the point distribution with a weight equal to the number of points on the envelop. This approach gives excellent results and returns the exact solution if the points were originally distributed on an ellipsoid (see Figure 4.3 for a 2D illustration).



(a) General 2D curve fit (b) Exact ellipse is recovered

Figure 4.3: Example fit of a 2D ellipse using PCA

"Is-inside" test

Performing the rasterization of the ellipsoid is significantly easier than rasterizing the original control volume. Ellipsoid principal directions are given by the eigenvectors of the correlation matrix:

$$R = V\Sigma V^T \tag{4.3}$$

with

$$V = [v_1, v_2, v_3] \in \mathbb{R}^{3\times 3} \tag{4.4}$$

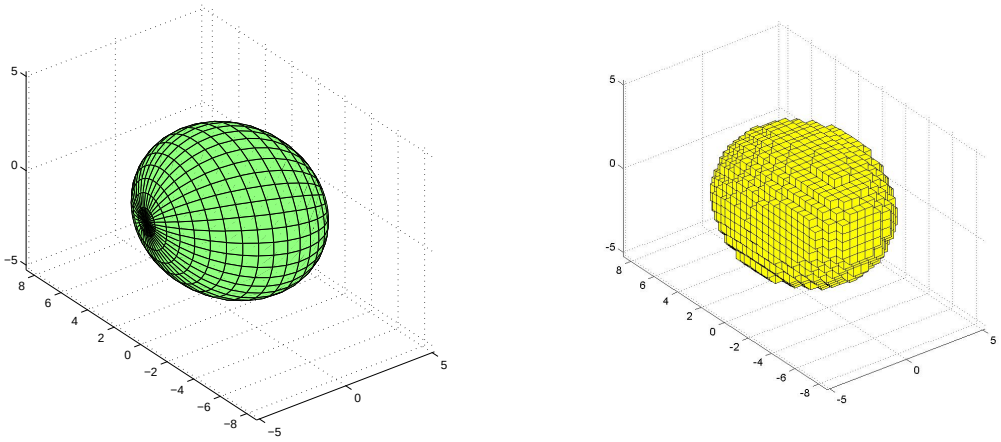
and

$$\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3) \in \mathbb{R}^{3\times 3} \tag{4.5}$$

The coordinate transform from rectangular to ellipsoid principal components is

$$x' = \Sigma^{-1} V^T x \tag{4.6}$$

where x is the column vector of the original rectangular coordinates and x' is the column



(a) Ellipsoid is fitted using principal component analysis

(b) Fine cells composing the approximated target region

Figure 4.4: Approximation of the target region

vector of the coordinates in the ellipsoid principal axes system.

The center point of each cell of the extended region is transformed into the ellipsoid principal-axes coordinate-system and the cell is considered to be inside the target region if

$$\left(\frac{x_1'}{a_1}\right)^2 + \left(\frac{x_2'}{a_2}\right)^2 + \left(\frac{x_3'}{a_3}\right)^2 \leq 1$$

where a_i ($i \in \{1, 2, 3\}$) are the ellipsoid axes lengths defined as $a_i = 1.96\sigma_i$. Figure 4.4 shows an example of a rasterization performed on a ellipsoid.

Extended region

Similarly to what was done in 2D, we define the extended region as the smallest rectangular subset of cells that includes all the tetrahedra connected to p. The extended region is easily identified on the fine grid by the sub-indice intervals $[i_{\min}, i_{\max}] \times [j_{\min}, j_{\max}] \times [k_{\min}, k_{\max}]$,

where for instance $i_{\min}$ is the minimum Cartesian ordering index of all the vertices connected to the node p via a tetrahedron edge.

4.1.2 Complications arising from multipoint flux approximation

The MPFA transmissibility calculations proposed by Verma (1996) and summarized in Section 2.3 provide a rigorous handling of both the full tensor permeability and general grids. However, as will be demonstrated in Chapter 5, pressure equation discretization using MPFA leads, under some circumstances, to inaccurate control-volume flux values. This is due to the poor conditioning of the linear system of equations. In these cases, it is not possible to transport fluid via the streamline simulation or using traditional finite volume approximations. To circumvent these difficulties, we propose an approach for upscaling within the context of a two-point flux approximation for the discretization of the pressure equation. This approach is presented in the next section. The two-point transmissibility upscaling (which we designate T^*-TPFA) will be shown to be much more robust than the $\mathbf{k}^*$-MPFA described previously, while preserving (and often improving) the accuracy of the coarse scale solution.

4.2 Transmissibility upscaling

In the previous section, we considered a flow-based technique for upscaling the permeability tensor. An alternative to this permeability upscaling is to directly compute upscaled transmissibility. In this approach, the transmissibility coefficients between cells are directly derived from extended local fine scale calculations rather than being computed from the upscaled tensors (as described in Section 2.3.2). Transmissibility upscaling has been

investigated by many authors within local, extended local, global and quasi-global contexts (see Durlofsky (2003) for a review and Chen *et al.* (2003) for recent developments).

In this work, we focus on the development of an extended local transmissibility upscaling technique. The quantity upscaled is a scalar value representing the flow rate induced by a pressure drop imposed between two primal nodes across a dual surface. The method is referred to as a two-point flux approximation T^* upscaling. The influence of neighboring pressure nodes (present in the multiple-point approach) is therefore neglected. This approximation would be expected to be accurate if the numerical discretization minimizes multipoint contributions (such as in methods using a k-PEBI dual grid). In the context of general anisotropy, two-point T^* upscaling also proves to be a reasonable approximation, as we will show in Chapter 5.

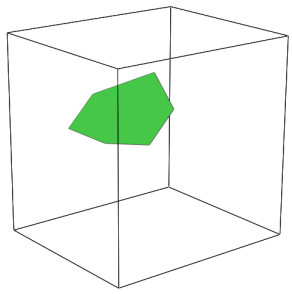
4.2.1 Problem definition for T^* calculation

Let us consider an interior face of the dual grid and the two corresponding adjacent cells (as shown in Figure 4.5). The two cells are general polyhedra and the face is a general polygonal surface which, in the case of a CVFE dual grid for instance, is in general non-planar.

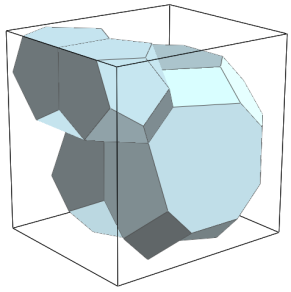
In TPFA, the flux across a face of the dual grid is written in terms of the difference between pressure values at the center of the dual cells (nodes of the primal grid). Denoting F_{ij} as the face between cells i and j, the flux q_{ij} across F_{ij} can be written as:

$$q_{ij} = -T_{ij}(P_j - P_i) \tag{4.7}$$

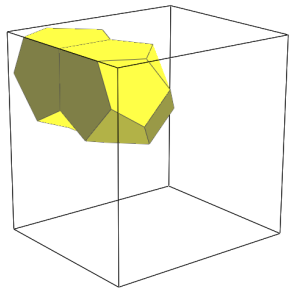
where T_{ij} is the two-point flux transmissibility coefficient.



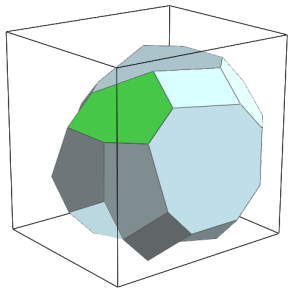
(a) Interior face of the dual



(b) Corresponding dual cells



(c) Left cell



(d) Right cell

Figure 4.5: Flow across a dual face involves the value of the pressure at the center of the two adjacent cells

A relatively simple approach is to attempt to directly approximate the permeability at the face as a function of the permeabilities at the points i and j. This approach does not account for anisotropy and can lead to significant inaccuracies. In the following section, we propose a more accurate approximation for T_{ij} which entails obtaining values for q_{ij}, P_i and P_j via fine scale flow simulations over the extended local domain containing cells i and j.

4.2.2 T^* calculation

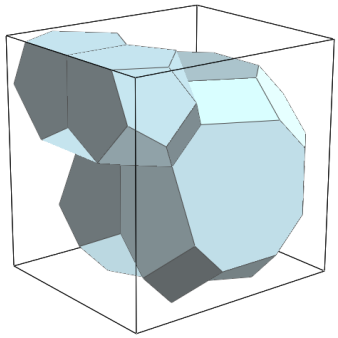
From a single phase flow calculation over an appropriate fine grid region, we can compute average values for the pressures in the cells i and j, designated $\langle P_i \rangle$ and $\langle P_j \rangle$. An average Darcy velocity $\mathbf{u}_{ij}$ can also be calculated in a volume including F_{ij}. Then, the average flux $\langle q_{ij} \rangle$ is obtained across F_{ij} in a straightforward manner, and the upscaled transmissibility is computed via:

$$T^*_{ij} = -\frac{\langle q_{ij} \rangle}{\langle P_i \rangle - \langle P_j \rangle} \tag{4.8}$$

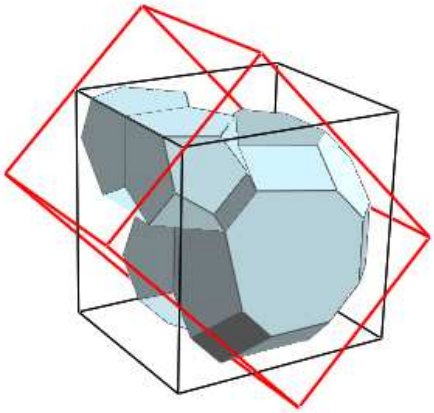
As in the case of permeability upscaling, the choices of the extended local domain, the type of boundary conditions used in the flow calculation and the averaging method must be addressed.

Extended region

Numerical tests showed that failing to align the extended region (and therefore the pressure gradient) with the direction of the line connecting the nodes noticeably reduced the accuracy of the calculation of T^*. Therefore the extended region must not only include the two relevant polyhedral cells associated with the transmissibility nodes, but must also be aligned with the vector connecting these two nodes (Figure 4.6).



(a) Region aligned with the Cartesian grid axes



(b) Region aligned with the line connecting the nodes

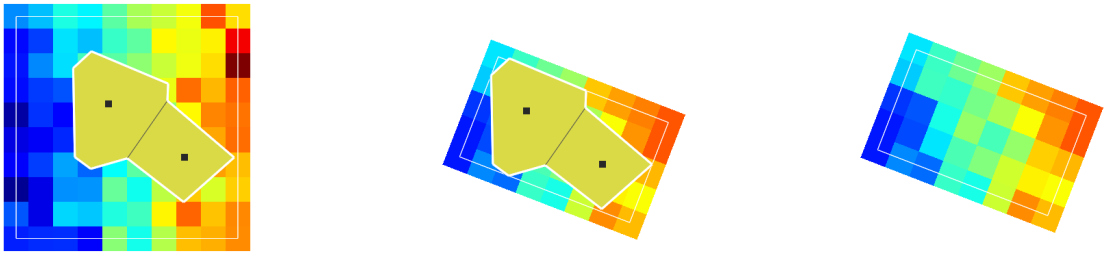
Figure 4.6: Aligning the extended region with the line connecting the nodes as in (b), proved to give superior results than simply using the alignment with the Cartesian grid axes as in (a)

The extended region is obtained by first determining the principal axes of the ellipsoid formed by the union of the two polyhedral cells. The permeability distribution is then transformed from the original Cartesian grid to the rotated system. As a consequence of this rotation, the volume support of the permeability distribution may be distorted and a resampling of the fine scale permeability may be necessary. Figures 4.7(a)-(c) show how a 2D extended domain containing the two target regions is extracted from the fine model. Figures 4.7(e)-(f) illustrate the approximation of the original permeability (shown in Figure 4.7(d)) using two different resampling frequencies. Section 5.2.3 demonstrates the benefit of using a resampling frequency greater than one in some cases.

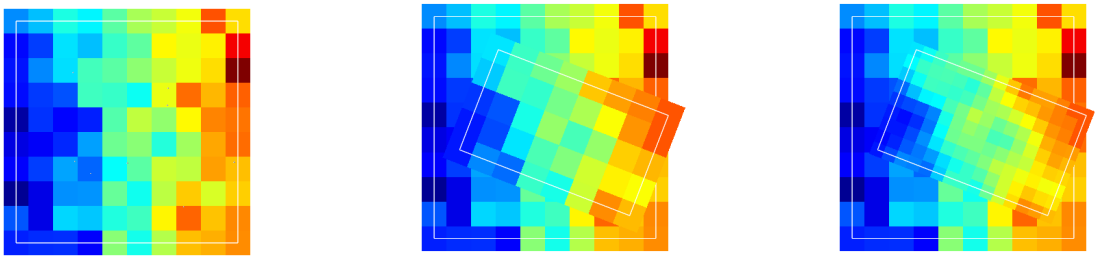
It should be noted that this permeability rotation can be performed without further calculations only when the background (geocellular) permeability is isotropic. If this is not the case, a tensor rotation must be applied to map the fine scale permeability to the rotated system.

Boundary conditions

Again, different boundary conditions can be employed to obtain the flow variables (pressure and velocity) needed to evaluate Equation 4.8 on the rotated system. The principal flow direction is determined based on the principal axes of the extended region (which is aligned with the segment connecting i to j). In that direction, a periodic boundary condition (subject to a jump in pressure) is applied. In the two other directions, periodic boundary conditions (without pressure drop) are applied.



(a) Fine grid and dual cells (b) Extracted extended region (c) Sampled permeability



(d) Full domain (e) Resampling frequency is 1 (f) Resampling frequency is 2

Figure 4.7: The extended region is rotated to be aligned with the connection direction; different resamplings of the permeability may affect the T^* calculation

Averaging technique

Two types of averaged values need to be computed to determine the transmissibility: (a) average pressure over the polyhedral cells and (b) average flow rate across the polygonal interface.

We again introduce an approximation for the volume over which we wish to compute the average values. In Section 4.1, we discussed a technique to approximate $\langle \nabla P \rangle$ for polyhedral cells. The extension of this approach to the determination of $\langle P \rangle$ is straightforward. In order to approximate the average flux $\langle q_{ij} \rangle$ across the polygonal face, we first compute the average velocity vector $\langle \mathbf{u} \rangle$ for the fine cells representing the face. Then, we compute the corresponding flux across the face. If the face is planar, we simply have:

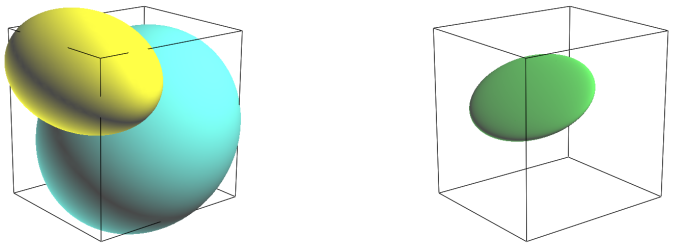
$$\langle q_{ij} \rangle = A_{F_{ij}} \langle \mathbf{u} \rangle \cdot \mathbf{n}_{F_{ij}} \tag{4.9}$$

where $A_{F_{ij}}$ is the area of the face connecting i and j. For the general case in which F_{ij} is not planar, it is decomposed into a piecewise planar representation. We then compute $\langle q_{ij} \rangle$ via

$$\langle q_{ij} \rangle = \left(\sum_{k=1}^{n_t} A_{F_{ij}}^k \mathbf{n}_{F_{ij}}^k \right) \cdot \langle \mathbf{u} \rangle \tag{4.10}$$

where n_t is the number of tetrahedra connected to the edge $[i, j]$.

The target region can either be approximated by a parallelepiped or by an ellipsoid as shown in Figure 4.8. The rasterization of the face is slightly complicated by the fact that the ellipsoid degenerates to an ellipse. In this work we chose to average velocity over the box fitting the face. Once the average pressures and flow rate are computed, T^* can be calculated using Equation 4.8.



(a) Average pressures are computed on the ellipsoid fitting the cell

(b) Average velocity is computed on the ellipsoid fitting the face

Figure 4.8: Target region approximated by an ellipsoid

4.2.3 Use of T^* in the streamline simulation

The discretization of the pressure equation is simplified by the use of TPFA. In particular, the preprocessing proposed by Verma (1996) is no longer required to derive flux continuous transmissibilities. Indeed, the transmissibility value derived from the T^* calculation can be readily input to the transmissibility matrix.

Nevertheless, for the purpose of tracing streamlines, we need to recover fluxes on the faces of the CVFE grid. The TPFA only provides the 'total' flux across the faces connecting two given primal nodes. The ambiguity of the definition of the flux on each individual face is resolved by apportioning the total flux based on the face area. It should be noted that a simplified streamline tracing method may result from the use of the two-point flux approximation. Such simplifications were, however, not investigated.

4.2.4 Advantage over MPFA

The validation of the T^*-TPFA is presented Chapter 5. In Section 5.2 it is shown that the T^*-TPFA compares favorably with flow based MPFA (using $\mathbf{k}^*$ upscaling) in cases for which MPFA works. The present section investigates additional advantages of the TPFA

over MPFA, particularly regarding the problem-matrix conditioning number.

Transmissibility matrix and condition number

Once the matrix is formed for the single phase incompressible problem (before any boundary conditions are applied), the matrix conditioning can be studied. The condition number reflects how near to singular the matrix is. It is also a measure of the sensitivity of the solution to perturbations of the matrix or right hand side coefficients. For a nearly singular matrix, direct methods generally lead to zero pivots and overwhelming numerical errors. In the case of iterative methods, the solution may not converge at all.

Strongly diagonal-dominant matrices usually lead to better condition numbers. Because MPFA tends to weaken diagonal dominance, putting more weight on off-diagonal terms, the condition number of matrices obtained using MPFA is generally greater than that of matrices obtained using TPFA discretizations.

Table 4.1 shows condition number values for different grids and transmissibility approximations. The cell shape is fixed and dimensions in the x and y directions are approximately 5 times greater than in the z direction. Two fine (reference) permeability distributions are considered, a homogeneous medium ($k = 10\,\text{md}$), and a heterogeneous model described in Section 6.2. Both models are comprised of $200 \times 100 \times 50$ cubic Cartesian cells. Two types of grids are considered, a uniform coarse tetrahedral grid containing 726 nodes and an adapted grid containing 468 nodes and obtained using fine scale flow rate information. Two different transmissibility approximations are considered: (a) the flow based upscaled TPFA (T^*-TPFA), and (b) the flow based upscaled MPFA ($\mathbf{k}^*$-MPFA). The condition number of the problem matrix is calculated for all of the cases.

		Permeability distribution	
		Homogeneous	Heterogeneous
Discretization type	$\mathbf{k}^*$ MPFA	1.4×10^3	1.3×10^5
	T^* TPFA	1.1×10^3	1.7×10^4

Table 4.1: Condition number for different grid types and transmissibility approximations

Results show that the condition number of the homogeneous and heterogeneous problems considered are clearly reduced using TPFA compared to MPFA. Further investigation is however required to determine why the MPFA method fails to provide acceptable results in some cases. For comparisons of flow results between T^*-TPFA and $\mathbf{k}^*$-MPFA, refer to Section 5.2

4.3 Grid generation tool

4.3.1 Gocad development platform

We now describe how the techniques discussed above were implemented into a research software platform that aims at investigating problems related to adapted grid generation. This platform, called StanLab, is based on a software development kernel and other tools provided by the Gocad research consortium. The Gocad consortium[2] is an academic consortium whose goal is to develop new computer-aided approaches for the modeling of geological objects. Key interests of the consortium include structural modeling, reservoir modeling and topology. The consortium produced a geomodeler called Gocad, now commercialized by a third party company[3]. The Gocad software serves as a licensed platform for all the research projects conducted within the Gocad consortium. Extensions to the

[2]http://gocad.ensg.inpl-nancy.fr
[3]http://www.earthdecision.com

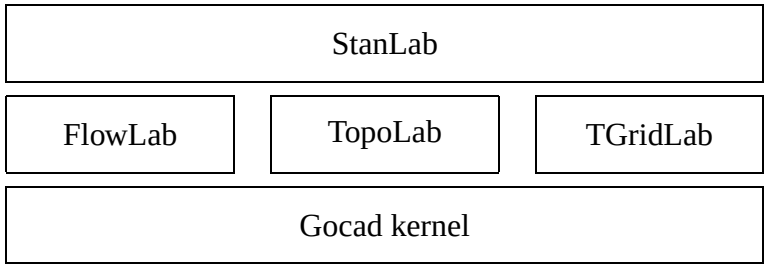


Figure 4.9: Organization of the Gocad plug-ins and plug-in developments

Gocad tools are called ‘plug-ins’ and can be independently developed by companies or universities. The following key plug-ins are used in the present work:

TopoLab, used for the construction of general polyhedral meshes. Allows the user to construct, edit and visualize complex cells such as control volumes as well as their associated properties. TopoLab was developed by Levy (1999) at the LIAD.

TGridLab, used for the generation of 2D/3D simplex meshes from structural information and resolution constraints. This is the principal tool we use for grid adaptation. TGridLab was developed by Lepage (2003) at the LIAD.

The hierarchical structure of the plug-ins allows for the development of new functionalities on top of existing plug-ins. In this work, the following plug-ins were developed (see Figure 4.9):

FlowLab, enables flow simulation for unstructured grids. FlowLab allows us to perform two phase flow simulation as well as fast streamline simulation for tracer flow. FlowLab is partially based on the unstructured black oil simulator Flex developed by Verma (1996). The FlowLab functionalities available through the StanLab plugin include MPFA transmissibility calculations, unstructured streamline or black-oil simulations.

StanLab, generates grids constrained to flow response. It includes the creation of flow information maps, flow-based upscaling, flow diagnostics using streamline simulation on both structured and unstructured grids (via FlowLab) and adapted gridding procedures via TGridLab. StanLab also provides an interface for all of the plugins mentioned above in one single convenient tool.

A more detailed description of the StanLab software is presented in the Appendix.

4.3.2 Constrained gridding algorithm

We now describe the gridding algorithms used to construct 3D reservoir models. The typical steps in the model construction are:

1. Construct a structural model from faults, horizons and other geological data. Define contacts between surfaces and surface intersections.

2. Construct a grid model that defines the topological constraints of the grid elements (tetrahedra). Examples of these constraints are:

   - Cell faces to conform to surfaces
   - Cell edges to conform to lines

3. Generate information maps via single phase or tracer flow calculations.

4. Incorporate the grid geometrical constraints such as:

   - Cell-shape quality control
   - Cell size constraints from spatial maps

5. Construct the grid, honoring the topological and geometrical constraints.

In our approach, the initial steps of the reservoir construction are performed using the Gocad structural modeling tools. Then, the topological constraints derived from the structural model are embedded into a macro-topological framework using TGridLab. This model is called a *wire frame model* (see Lepage, 2002). The wire frame model (WFM) is the starting point for the domain tesselation. Flow information maps (obtained via StanLab) are included in the WFM where they serve as grid resolution constraints. The grid construction proceeds as follows:

1. Construct an initial discretization of all the lines of the model. These lines are topologically connected line constraints that may be the result of the intersection of fault and horizon constraints.

2. Construct a conforming triangulation of all the surfaces:

   - Obtain the initial constrained grid using a *conforming* approach, allowing the insertion of Steiner points on the constraint lines (see Figure 4.10).
   - Apply Delaunay refinement algorithm to account for shape quality criteria as well as grid resolution constraints on the surfaces.

3. Construct a conforming tetrahedral grid. The same two steps as described above are applied to the tetrahedral grid:

   - Obtain an initial conforming grid (Steiner points on constraint surfaces).
   - Apply Delaunay refinement algorithm.

The Delaunay refinement algorithm of Lepage (2002) follows the algorithmic framework of Shewchuk (1998). Extensions introduced by Lepage (2002) include the ability to

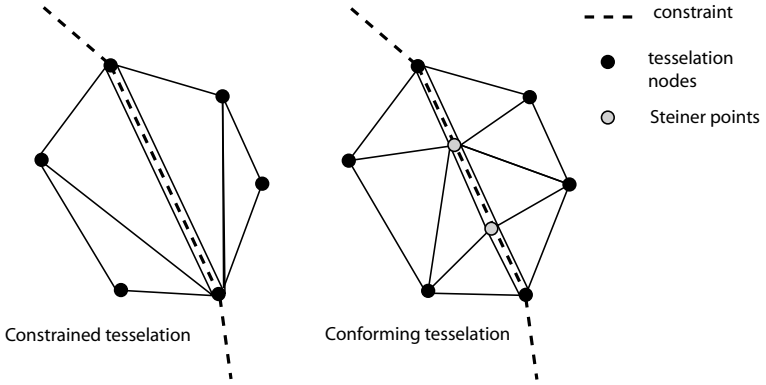


Figure 4.10: Two approaches to Delaunay tesselation: constrained and conforming. The conforming approach may require the insertion of Steiner points, though it preserves the Delaunay criteria globally

handle curvilinear boundaries and to impose grid resolution constraints anywhere in the mesh. TGridLab enables the user to take into account any number of resolution constraints in combination with a constraint on the shape of the elements.

4.4 General methodology

The grid generation may be viewed as an iterative process in which gridding parameters are estimated, tested, and updated until a satisfactory flow diagnostic is obtained. Each cycle of the grid generation relies on three key stages: (a) generate/adapt the unstructured grid, (b) calculate effective (upscaled) properties on that grid, (c) obtain the flow responses of the coarse model and compare with reference solutions for selected boundary conditions. Reference flow responses for the problems included in the diagnostic, and any fine flow information needed in the grid adaptation, must be computed prior to the grid adaptation process. We first review the gridding capabilities of TGridLab. Next we detail the formulation of the gridding parameters and their selection, and describe how they determine the adapted grid.

4.4.1 Grid generation/adaptation

Among other functionalities, TGridLab allows for the construction of constrained tetrahedral grids, using geometrical constraints (such as faults, boundaries and lines), and grid-resolution constraints.

As indicated above, the construction of an adapted grid is divided into two basic steps: (a) build an initial tesselation that honors a collection of geometrical constraints, (b) iteratively insert points in the tesselation and update the topology. The point insertion is done according to a grid quality criterion that can either be related to the shape or the size of the tetrahedral elements. The tesselation is modified by insertion of points at locations that improve the quality of the cells. This requires the definition of a cell quality function and a global threshold below which tetrahedra are subject to an update. The update consists of a point insertion at the tetrahedron circumcenter and a local modification of the grid topology that enforces the Delaunay criterion while also preserving the geometrical constraints. After each iteration, the quality of the cells influenced by the insertion is recomputed. The iterations are performed until the quality of all tetrahedra is above the specified threshold.

4.4.2 Resolution constraint

Definition

As mentioned previously, the quality of a tetrahedron can be a function of its shape and/or size. The case of grid adaptation using a shape quality criterion is not investigated in this work; for further details on this approach, refer to Lepage (2002). In the case of a size quality criterion, the quality of the tetrahedron is determined by its size and the value of a

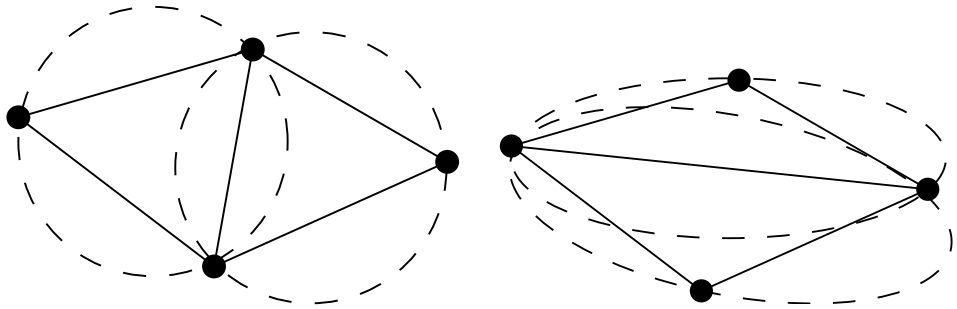
resolution constraint. The resolution constraint determines the maximum size of any tetrahedra in the grid. The size of a tetrahedron is defined as the radius of its circumscribed sphere. The resolution constraint can either be a fixed value for the entire domain, or, more commonly, a spatially varying quantity. The size quality of any tetrahedron is therefore computed as the ratio of the value of the resolution constraint in the region in which the tetrahedron lies to the radius of its circumscribed sphere. If the ratio is smaller than a specified threshold, the tetrahedron is subject to an update (as explained in the next subsection).

Resolution constraint and grid anisotropy

A noticeable restriction of this procedure is that it forces the grid being constructed to satisfy the Delaunay criterion in a Euclidean sense. This limits the amount of geometrical anisotropy the grid can have because the Delaunay "empty sphere" property forces connections to closest neighbors (Figure 4.11). This limitation could be circumvented by evaluating the size quality criteria in a "metric" sense instead of a Euclidean sense. The sphere radius would then be evaluated in a transformed space. This would require the use of a (potentially spatially varying) tensor field serving as the metric. The capability of imposing geometrical anisotropy along these lines is not an available feature of the current version of TGridLab. We will see in Section 6.2 that some grid anisotropy can still be obtained, which allows us to take advantage of the layered character of a reservoir model.

Resolution constraint and information maps

The resolution constraint can be viewed as a 3D map that we use to impose a given grid size at different locations of the model. As we will see in Chapter 6, some benefit may result from using fine scale flow information as a resolution constraint. The principal challenge



(a) Delaunay criterion in the Euclidian space

(b) Delaunay criterion in a metric space

Figure 4.11: Enforcing the Delaunay criterion in a metric space (b) rather than in the Euclidean space (a) allows the user to impose anisotropy to the grid

in grid adaptation using a resolution constraint is the optimal construction of the constraint.

In reservoir simulation, the main benefit expected from grid adaptation is the reduction of the number of unknowns in the discretized flow equations, as this number is a linear function of the number of nodes (dual cells) in the grid. Grid adaptation allows us to use a given number of nodes (unknowns) more efficiently. The efficiency is measured in terms of accuracy of the flow calculations obtained on the adapted grid with respect to reference solutions obtained on the fine model.

The fine solution is usually not available, hence the necessity of constructing the resolution constraint using information that (a) is affordable so it can be obtained on the fine grid, and (b) is meaningful to the physics of the problem that will be modeled on the adapted grid. We suggest in this work the derivation of information maps from tracer flow simulations on the fine model. These simulations are often affordable since they require only one pressure solution (for a given set of boundary conditions) and can provide single phase as well as displacement information such as total flow rate and time of flight.

4.4.3 Information maps

We now discuss two types of flow information provided by tracer flow simulations that can be used to construct a resolution constraint. These property fields, called flow-rate and breakthrough-time maps, were already introduced in a 2D context in Chapter 2. They are reintroduced here for completeness as their computation can now be accomplished though the gridding tool that we developed. Flow information maps are applied to the simulations presented in Chapter 6.

Mean flow-rate map

We obtain a mean flow-rate map in a straightforward manner. It is the weighted average of the local velocity norm obtained from one or more single phase flow simulations. The simulations differ based on the boundary conditions applied. For instance we may consider three face-to-face flow simulations (left-to-right, front-to-back and top-to-bottom) with fixed pressures so that the average pressure gradient is equal to 1 in all solutions. As a consequence, the flow rates calculated are "normalized" for a unit pressure gradient. This is achieved for instance on a 1D reservoir of length L by applying fixed pressures on each side with values P_0+L and P_0 respectively. Other boundary conditions and other averaging methods may be considered. For instance some special well configuration may be of interest. Also, assigning a larger weight to flow rates in the z direction may enhance vertical flow connection features, which will then be better captured by the resolution constraint.

Breakthrough-time map

The breakthrough time map draws a correspondence between time values on the PVI axis and regions of the reservoir that are completely swept by the injected fluid at that time.

The map is obtained by assigning to each streamline the value of the time at which a particle tracked on that streamline would break through. For each cell of the reservoir, a breakthrough time value is calculated by the weighted average of the values carried by the streamlines crossing the cell. Enforcing that all cells be traversed by at least one streamline requires that many streamlines be traced. Instead of this, we trace a fixed number of streamlines from the injector to the producer, then trace other streamlines from the producer back to the injector. The breakthrough time values at cells that are not traversed are estimated through a kriging technique.

The breakthrough time map is meant to be used in the resolution constraint as a way to capture important events in the fractional flow curve (such as water breakthrough) and to introduce grid refinement in appropriate regions of the model.

4.4.4 Grid adaptation parameters

We now discuss our choices for the adapted gridding parameters. The grid adaptation is basically determined by the choice of three elements: (a) the flow-information maps, (b) the resolution constraint (which transforms the flow information into a target grid size), and (c) the flow diagnostic which assesses the quality of the grid.

Resolution constraint

Let us consider the case of grid adaptation using only one flow-information map. The property at each fine scale Cartesian cell is mapped to a point target grid resolution (size) as shown in Figure 4.12. The a and b parameters relate to what proportion of the domain will be refined. Figure 4.13 shows that a and b can be chosen so that P_a percent of the information map specifies a grid size of at most s_a and $(100 - P_b)$ percent specifies a grid

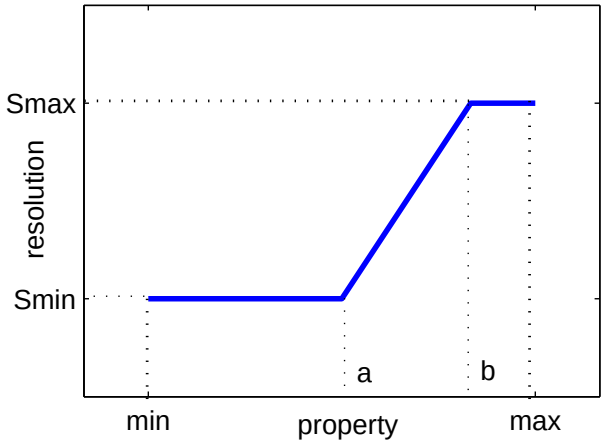


Figure 4.12: Function transforming the property map into a pointwise target resolution

size of at most s_b. Assuming the shape of the resolution function is fixed as shown in Figure 4.12, we then have four parameters (P_a, P_b, s_a and s_b) for the grid adaptation.

The criterion used to determine the target size of a tetrahedron can be obtained using any of the following procedures:

- Look-up the point value of the grid resolution at the circumcenter of the tetrahedron
- Average the point values over the volume spanned by the sphere circumscribed to the tetrahedron
- Select the minimum value of the point grid resolution over the volume spanned by the sphere circumscribed to the tetrahedron

For continuous information maps such as mean flow rates, it is preferable to use the minimum criterion as the other criteria may not allow us to capture thin coherent structures from the information map.

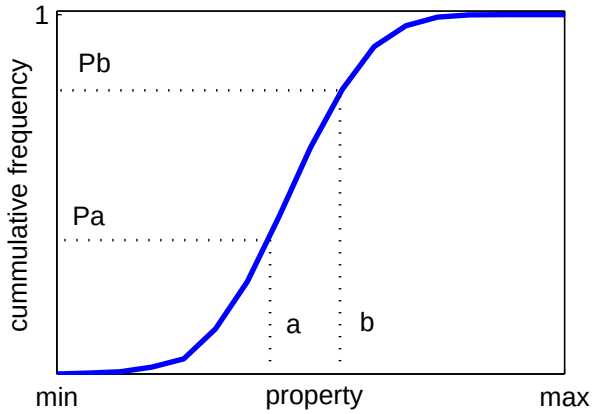


Figure 4.13: Parameters a and b are used to specify which proportion of the domain is to be refined

Small-to-large grid size ratio

Instead of using the values of s_a and s_b to characterize the resolution constraint, we prefer to use the ratio s_a/s_b and N, the number of nodes in the grid. We call s_a/s_b the small-to-large grid size ratio. For a given small-to-large ratio, we can select different pairs of (s_a, s_b) values, resulting in different adapted grids of size N (function of s_a and s_b).

Aspect ratio

We define the grid anisotropy of a tetrahedral cell as the aspect ratio of the box circumscribed to the tetrahedron vertices and aligned with the coordinate axes. The aspect ratio is specified via $a_x \times a_y \times a_z$ and is defined up to a multiplicative constant. For instance, a $10 \times 10 \times 2$ aspect ratio is equivalent to $5 \times 5 \times 1$ and expresses that the size of a cell in the x and y directions is 5 times greater than in the z direction.

Reservoir property models often exhibit large degrees of statistical anisotropy. However, the current functionalities of TGridLab (discussed in Section 4.4.1) do not permit us to constrain the grids to approximately reproduce a user defined anisotropy. Nevertheless,

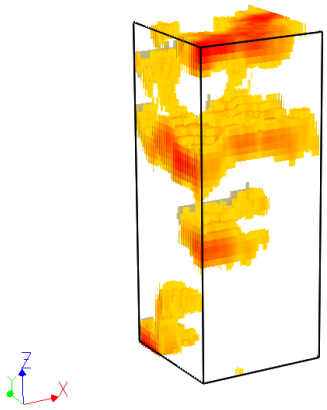
some grid anisotropy can be obtained by performing the grid adaptation in a transformed space.

In order to introduce grid anisotropy in the tetrahedral elements, we modify the space in which the Delaunay criterion is evaluated. This is achieved by stretching the coordinate axes of the fine model containing the resolution constraint information. Once the adapted grid is obtained, the coordinates of the grid nodes are mapped back into the original coordinate system without modifying the topology. The grid is therefore generally no longer Delaunay in physical space. Figure 4.14 illustrates the transformation steps. Note that flow related equivalent quantities such as permeability and transmissibility are computed in the original (physical) coordinate system.

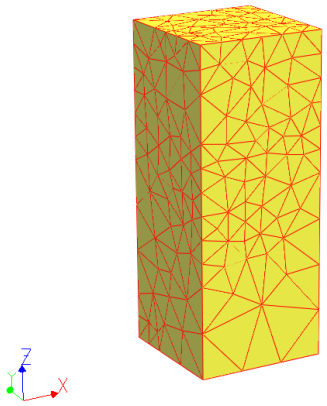
Flow diagnostic

The flow diagnostic is defined as the quantitative level of agreement between the flow results obtained on the coarse and fine (reference) grids. The flow diagnostic may involve one or more flow scenarios to ensure some level of robustness of the grid to changes in boundary conditions. A flow scenario is defined by the type of displacement modeled and the boundary conditions applied in the simulation. In this work, we only consider tracer flow to assess the flow response of a model, as it can be obtained in an efficient manner for both the fine and coarse models. The type of boundary conditions applied are 'face-to-face' fixed pressure conditions. Pressure is fixed on opposite faces of the model and no-flow conditions are applied on all other boundaries. The methodology however is in no way restricted to this particular choice of boundary conditions.

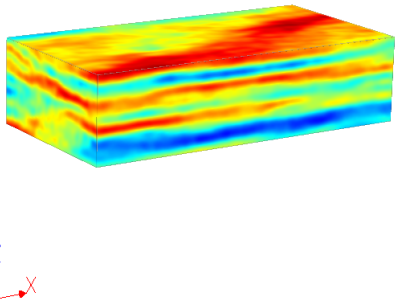
The quantities chosen for comparing the tracer flow simulations are: (a) the total flow rate observed across the model and (b) the L_1 norm of the error in the fractional flow curve



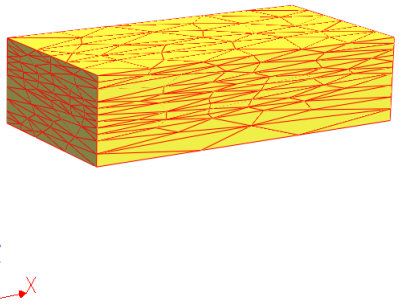
(a) Cartesian grid in stretched space (target grid resolution)



(b) Tetrahedral grid in stretched space, adapted using the resolution constraint in (a)



(c) Cartesian grid in original space (target grid resolution)



(d) Tetrahedral grid in original space

Figure 4.14: The target anisotropy ratio is introduced by stretching the physical space and then performing an isotropic meshing in the transformed space

taken between 0 and 1 PVI. The error introduced by the coarsening is therefore measured by the pair:

$$\text{error}(\text{grid}, \text{flow problem}) = (Q_{\text{err}}, L_{1\text{err}}) \tag{4.11}$$

with:

$$Q_{\text{err}} = \frac{(Q/\Delta P)^c}{(Q/\Delta P)^f} \quad \text{and} \quad L_{1\text{err}} = \frac{\int_0^1 |F_w^c - F_w^f| \, d\tilde{t}}{\int_0^1 |F_w^f| \, d\tilde{t}} \tag{4.12}$$

where F_w designates the water cut as a function of PVI ($\tilde{t}$) and the f and c superscripts refer respectively to fine and coarse quantities.

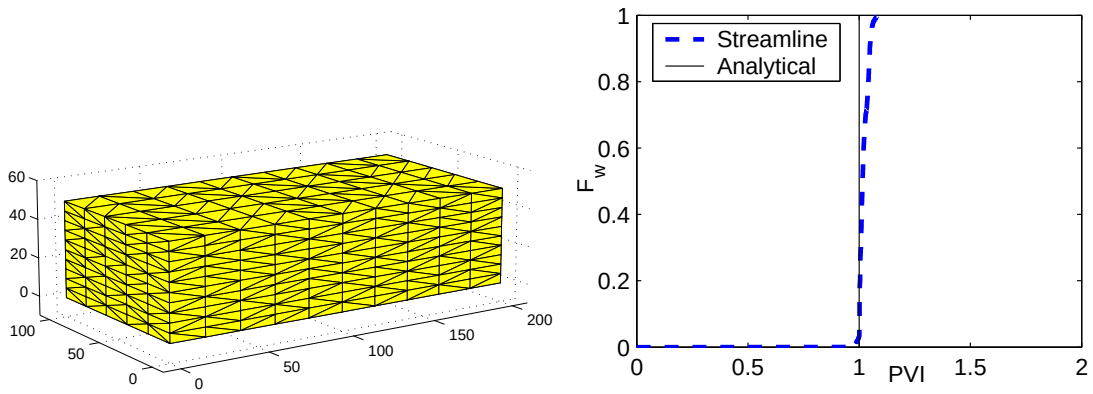
Chapter 5

Validation of methods

This chapter presents validation tests for the techniques developed in Chapters 3 and 4. The first section demonstrates the accuracy of the unstructured grid streamline simulator. We next illustrate the accuracy of the T^*-TPFA and its superior robustness compared to $\mathbf{k}^*$-MPFA for heterogeneous cases. We also include a brief discussion on why and when MPFA may fail to provide accurate fluxes for streamline tracing and transport modeling.

5.1 Streamline method

This section tests the streamline method described in Chapter 3 on 3D unstructured grids. The first series of tests demonstrates that the method correctly reproduces analytical solutions for simple problems. We then provide comparisons between the streamline method and established techniques for more complex problems.



(a) 726 node uniform tetrahedral grid (b) Water cut comparison

Figure 5.1: Streamline simulation on a homogeneous isotropic model

5.1.1 Homogeneous model

We investigate here the flow results produced by the streamline simulator for a homogeneous case. The test case used is a fully 3D unstructured model with homogeneous isotropic permeability. Boundary conditions were prescribed to give a 1D flow. The unstructured grid used for the streamline simulation is shown in Figure 5.1(a). Two fixed pressures were applied on the left and right faces of the model to produce a flow in the x direction. The total flow rate in this case is given by:

$$Q = \frac{k\Delta P L_y L_z}{\mu L_x}, \tag{5.1}$$

where L_x, L_y, L_z are the dimensions of the model, k is the permeability and μ is the viscosity of the fluid displaced. Breakthrough of injected fluid should occur at $\tilde{t} = 1$ PVI. The analytical total flow rate was recovered exactly by the unstructured pressure solver and the streamline tracing reproduced the expected water cut curve very closely, as shown in Figure 5.1(b).

5.1.2 Layered model

We now test our streamline method on a 3D layered case. The model is composed of four layers of equal thickness as shown in Figure 5.2(a). Because of the general shape of the unstructured dual grid, the permeability distribution is only approximately captured by the unstructured model (as shown in Figure 5.3(a)). The boundary conditions considered are fixed pressure on opposite faces, creating a linear flow parallel to the layers. The total flow rate is given by:

$$Q = \frac{\overline{k}\Delta P A}{\mu L}, \tag{5.2}$$

where $\overline{k}$ is the arithmetic mean of the layer permeabilities, L the length of the model, and A is its cross sectional area. The fraction of the total flow in each layer i is given by:

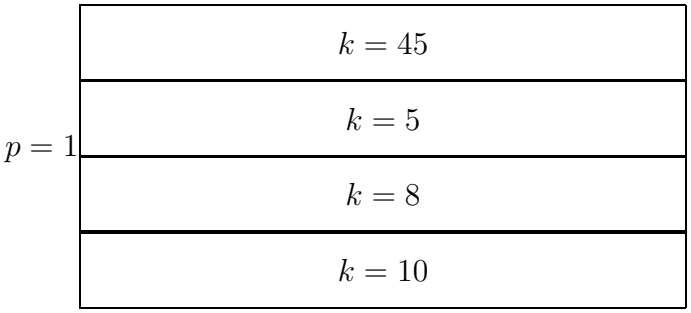
$$\frac{q^i}{Q} = \frac{1}{4}\frac{k^i}{\overline{k}}. \tag{5.3}$$

The breakthrough time of each layer is given by:

$$\mathrm{BT}^i = \overline{k}/k^i. \tag{5.4}$$

The oil cut comparison is shown in Figure 5.3(b). The curve exhibits the correct breakthrough time and overall shape but does not reproduce the sharp piston-like effect of the analytical solution. This is due to the fact that the layer geometry is only approximate, as we now demonstrate.

When considering a series of unstructured grids with increasing resolution, the representation of the layer geometry improves, as shown in Figure 5.4. The flow results converge

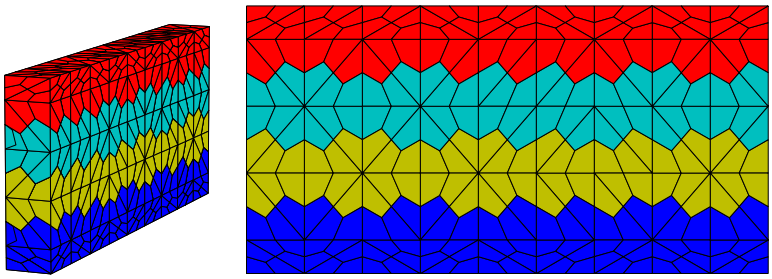


(a) Layered model

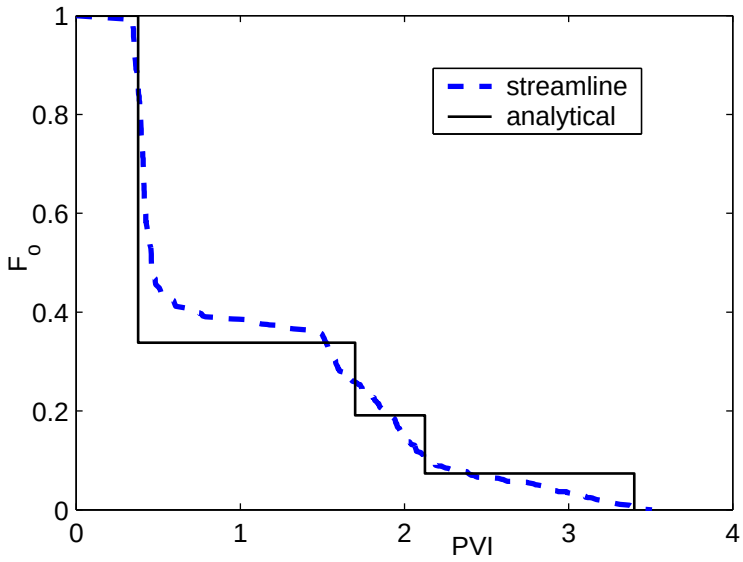
i	k^i	q^i/Q	BT^i
1	45	0.662	0.378
2	5	0.073	3.4
3	8	0.112	2.125
4	10	0.147	1.70

(b) Analytical results

Figure 5.2: Layered model, analytical rates and breakthrough times



(a) Side views of the unstructured dual grid (note the approximate geometry)



(b) Oil-cut

Figure 5.3: Grid and flow results on the unstructured grid

toward the analytical solution both in terms of oil-cut and total flow rate, as shown in Figure 5.5.

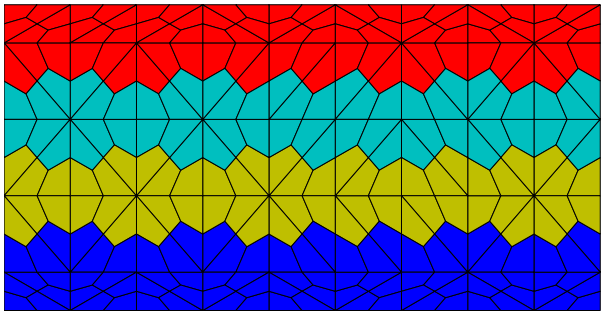
5.1.3 3D flow in a homogeneous cube

We now consider a cubic model with a homogeneous isotropic permeability. The unstructured streamline simulator is tested against a commercial Cartesian streamline simulator (3dsl). Fixed pressures are applied at diagonally opposite corners and tracer flow is again considered. The grid used for the unstructured simulation contains 1728 nodes and is shown in Figure 5.6. The flow result is shown in Figure 5.7. Agreement between the Cartesian streamline simulator and our unstructured simulator is excellent. For comparison, the solution obtained using a standard finite difference simulator with upstream weighting is also shown on the plot[1]. The finite difference result shows an earlier breakthrough, which is likely due to the effects of numerical diffusion. These effects are not present in the streamline simulations.

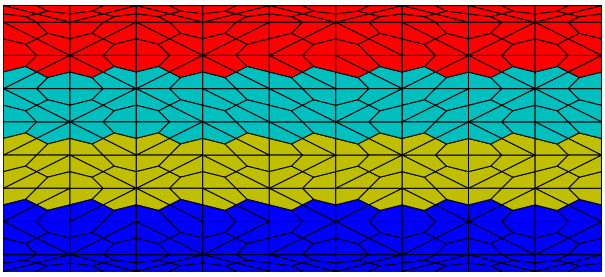
5.1.4 General heterogeneous test

We finally consider a general heterogeneous case. The model is a cube of side length L, populated with a log-normally distributed permeability field. The variogram model is statically isotropic with range $L/3$. The permeability distribution is shown in Figure 5.8(a). We consider a tracer simulation imposing fixed pressures at opposite corners of the model (as in the previous example). We wish to compare the flow results for the streamline simulation and a Cartesian finite difference simulation. A finer ($30 \times 30 \times 30$ cell) Cartesian grid was introduced for this comparison and the permeability distribution was resampled onto this

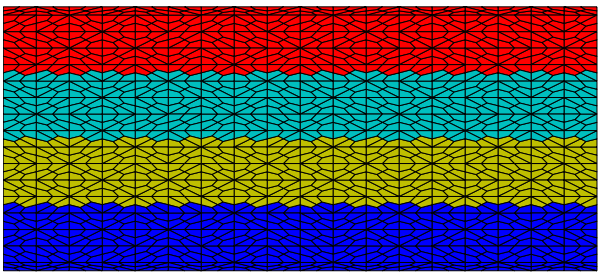
[1]The program used here is GPRS (Cao, 2002)



grid 1

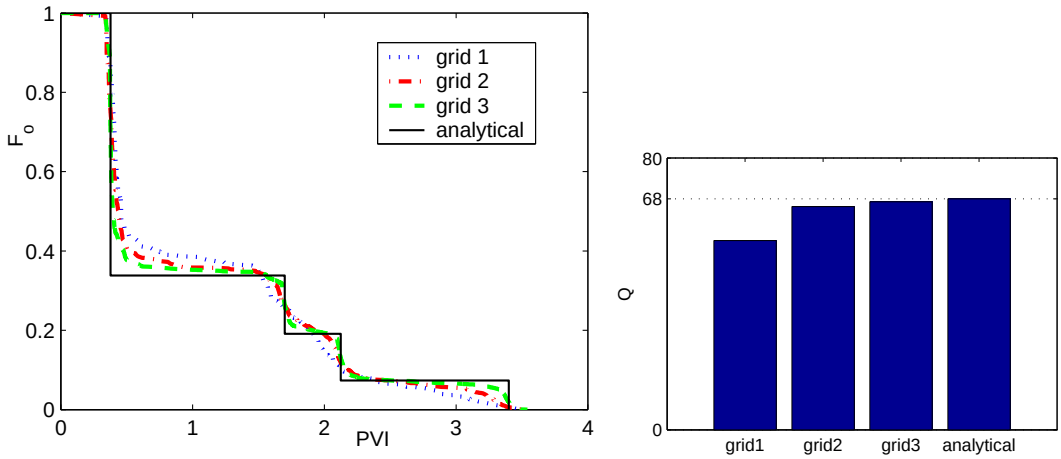


grid 2



grid 3

Figure 5.4: Different 3D grids used to approximate the layered mode (side views)



(a) Oil-cuts (b) Total flow rate

Figure 5.5: Flow results obtained using the grids shown in Figure 5.4

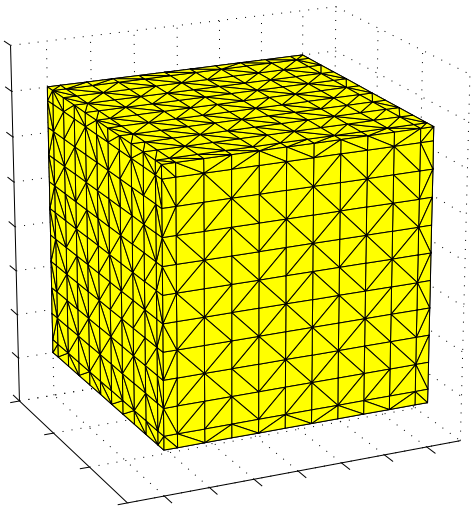


Figure 5.6: Grid used in the 3D comparison between unstructured and Cartesian streamline methods

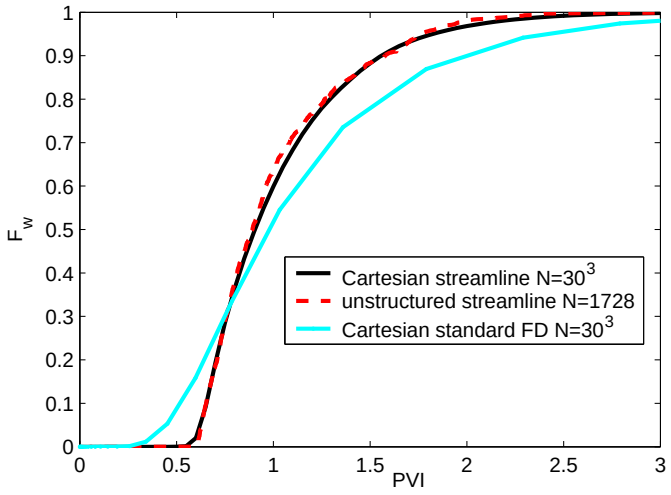
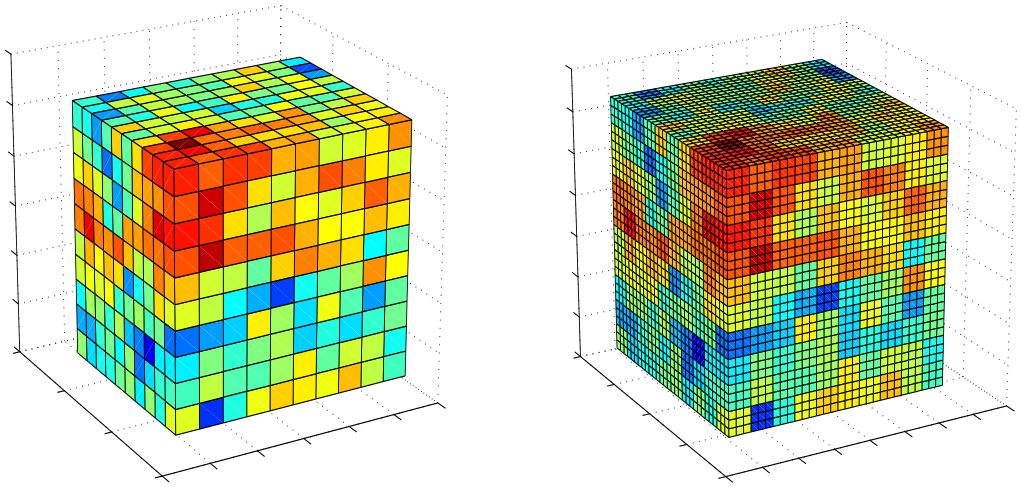


Figure 5.7: Comparison between unstructured and Cartesian streamline methods (permeability is homogeneous)

fine grid as shown in Figure 5.8(b). We considered a 1728-node unstructured grid to perform the streamline simulation. The pressure discretization used MPFA. The permeability model used for that case is shown on the dual grid in Figure 5.9.

The flow results are presented in Figure 5.10. The unstructured streamline solution now shows an earlier breakthrough time when compared to the Cartesian streamline simulation. This difference is likely caused by the difference in geometry of the permeability distribution (see Figures 5.8(a) and 5.9). The finite difference simulator again shows an earlier breakthrough presumably, as a result of numerical diffusion effects.

We now consider the permeability distribution shown in Figure 5.9 and compare the results obtained using our streamline simulator to those of GPRS. In this case, GPRS treats the system as unstructured (through the use of a connection list). The transmissibility coefficients are exactly the same in both simulations, only the method for the solution of transport differs. Figure 5.11 exhibits a reasonably close agreement between the two methods. Again, the finite volume results display the effects of numerical diffusion, which



(a) Coarse permeability model

(b) Permeability model used in the finite difference simulation

Figure 5.8: Permeability models used in the heterogeneous case comparing unstructured-streamline to finite difference simulation

are absent in the streamline simulation.

In this section, we demonstrated the accuracy of the unstructured 3D streamline simulator. Homogeneous models showed close agreement with analytical results and results from a commercial streamline simulator. For heterogeneous cases, the method exhibited good agreement, but only when the geometry of the heterogeneity was approximated with sufficient accuracy. This feature is not a consequence of the streamline method itself but is rather due to differences in the pressure solutions for different geometrical representations of the heterogeneity.

5.2 Comparison of TPFA and MPFA

For high grid aspect ratios (resulting in high grid anisotropy), and for high permeability anisotropy, tests showed that the MPFA discretization could give nonphysical results. For

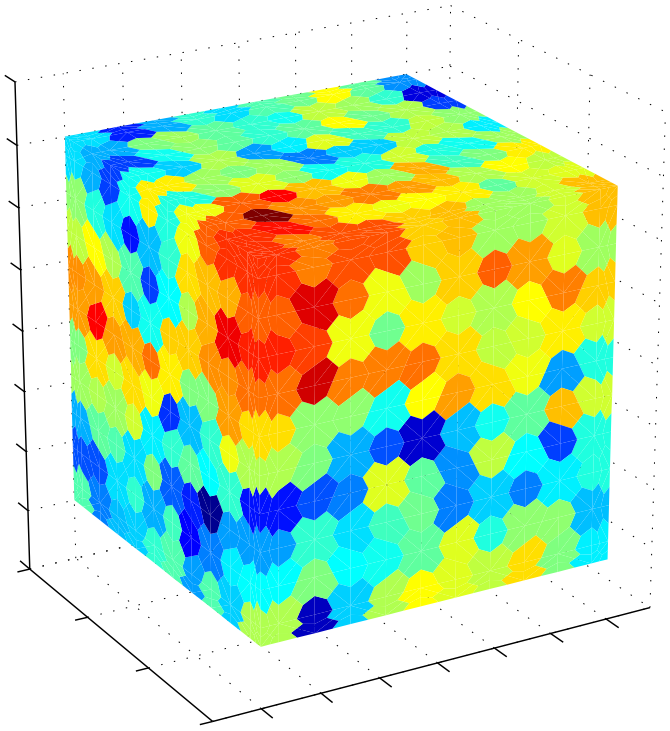


Figure 5.9: Permeability model used in the CVFE streamline simulation

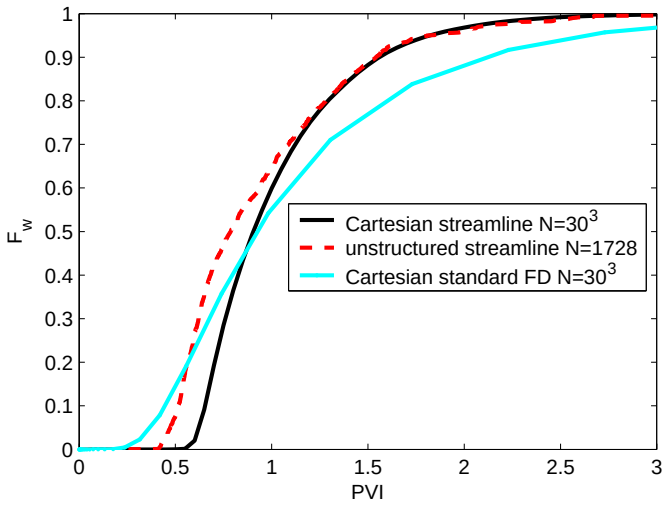


Figure 5.10: Flow results comparing the Cartesian-streamline, Cartesian-finite-difference and unstructured-streamline simulations (heterogeneous case)

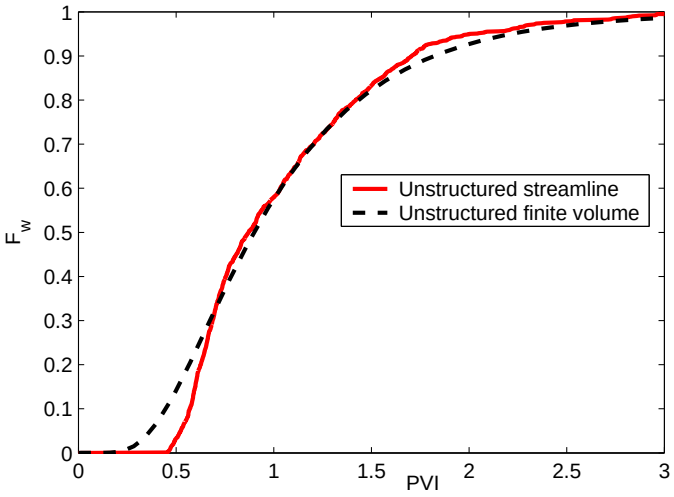


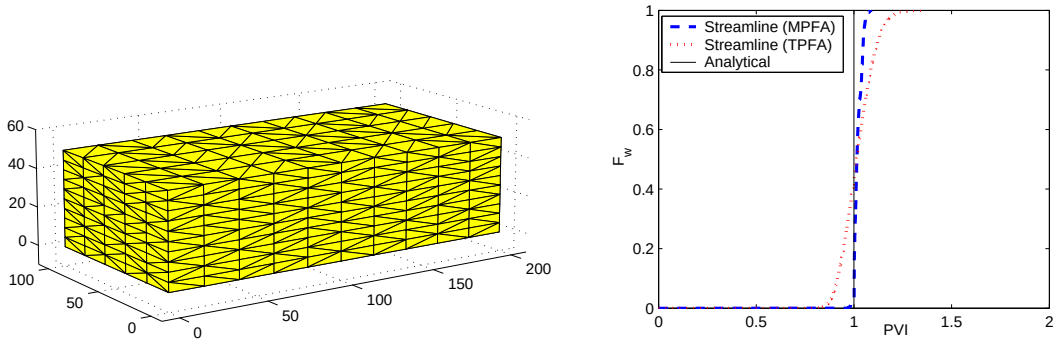
Figure 5.11: Flow results comparing the unstructured streamline method to finite volume simulation using the same grid (heterogeneous case)

example, for a horizontal-to-vertical aspect ratio of 10 to 1, streamline simulations performed on a homogeneous permeability field produced nonphysical swirling streamlines. In some heterogeneous cases, the problem matrix even became singular. These difficulties appear to result from high degrees of anisotropy. In such cases, the contribution from a point k may dominate the flux from cell i to j. This is, however, a tentative explanation; more investigation is required to determine practical criteria for the use of MPFA. It is also possible that alternate MPFA stencils may provide more robust results.

Due to the occasional failure of MPFA, we designed an alternate upscaling technique that provides coefficients for a two-point flux approximation as described in Section 4.2. We now compare flow responses using the two-point fluxes (T^*-TPFA) and the multiple-point fluxes ($\mathbf{k}^*$-MPFA) for both homogenous and heterogeneous systems.

5.2.1 Comparison for a homogeneous medium

This case compares TPFA and MPFA approximations for a homogeneous system. The model is a parallelepiped with dimensions of $L_x = 1$, $L_y = 0.5$ and $L_z = 0.25$. The unstructured grid used in the comparison is shown in Figure 5.12(a). In order to calculate two-point transmissibility coefficients, a fine Cartesian grid ($200 \times 100 \times 50$) was populated with uniform permeabilities and "upscaled" to the coarse tetrahedral grid using our T^* calculation. The permeability used in the MPFA calculations was set to be the known permeability and was not obtained via flow-based upscaling. A tracer flow simulation was performed using both the MPFA and the TPFA with the "upscaled" transmissibilities. Our streamline simulator was used in order to avoid numerical diffusion effects. A total of 726 nodes was used. Figure 5.12 shows the comparison of the simulated water cuts. These results show good agreement between MPFA and TPFA discretizations, though the TPFA



(a) Grid (b) water cut

Figure 5.12: Comparison of MPFA and TPFA for a homogeneous isotropic model (1D flow)

results are slightly more diffused (due to the "upscaling" step). The total flow rate mismatch observed was only about 5% for this case (other homogeneous simulations showed similar errors). This suggests that the two-point approximation may be adequate in terms of providing results in reasonable agreement with MPFA calculations.

5.2.2 Comparison for heterogeneous medium

We now consider more general permeability distributions where upscaling is performed before applying either TPFA or MPFA. The fine model is a $200 \times 100 \times 50$ Cartesian model with a layered, log-normally distributed permeability field. This model will be described in more detail in Section 6.2; it is shown in Figure 5.13. The comparison is performed considering fixed pressure boundary conditions on opposite faces of the model and no-flow and the other four faces. Two series of tests are conducted with the following boundary conditions: (1) left-to-right flow and (2) front-to-back flow. For each flow problem, we consider three different grid aspect ratios (as defined in Section 4.4) and two different coarse model sizes.

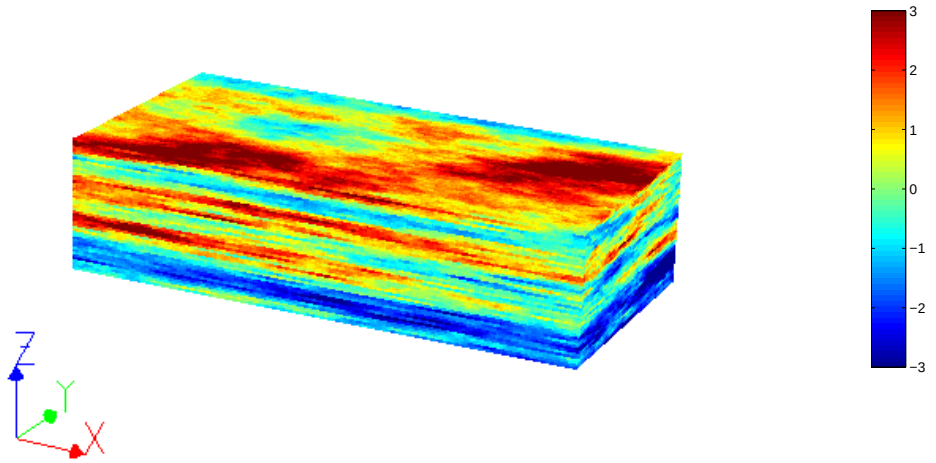
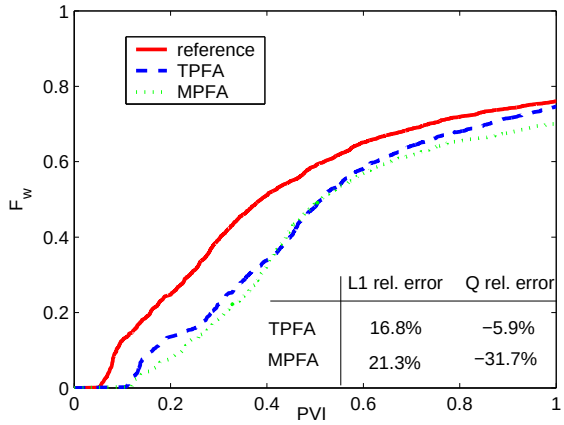


Figure 5.13: Layered reservoir, $\log k$

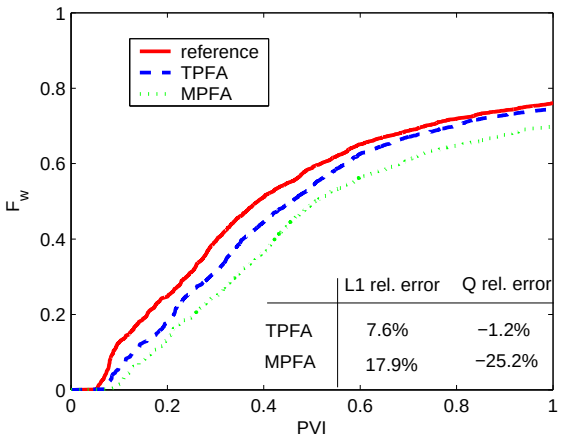
For the two flow problems, the fine Cartesian tracer flow response is calculated using a streamline simulator[2]. For each grid tested (6 in total), the unstructured streamline simulator provides the flow response for both the T^* and $\mathbf{k}^*$ flux approximations. The flow results are compared in terms of: (a) errors on the total $Q/\Delta P$ and (b) L_1 norm of the differences in fractional flow curves between 0 and 1 PVI (see Equation 4.12 and discussion in Section 4.4.4 for details). The coarse models tested represent an upscaling factor of approximately 1000. Figures 5.14 and 5.15 show the flow comparisons for both T^*-TPFA and $\mathbf{k}^*$-MPFA for the left-to-right and front-to-back flow problems respectively.

A first observation is that for a given number of nodes in the coarse model, some aspect ratios perform better than others (the aspect ratio $1 \times 1 \times 1$ is the worst). However, our primary concern here is not the absolute accuracy of the coarse flow results with respect

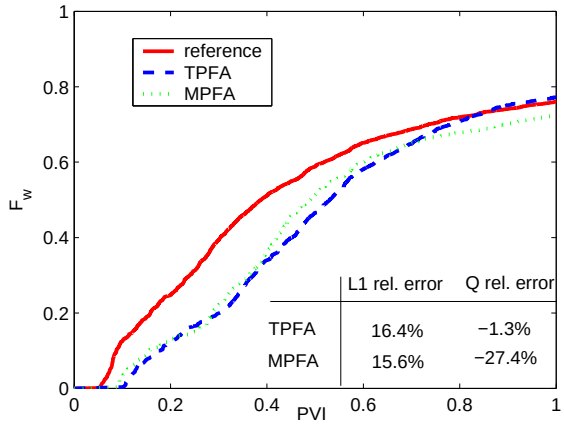
[2]implemented in the StanLab platform



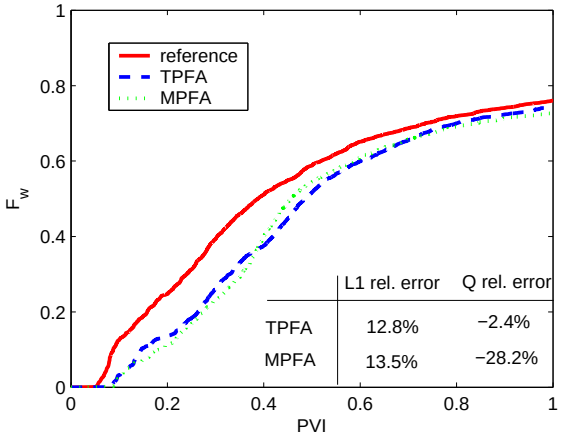
(a) $10 \times 5 \times 1$ (468 nodes)



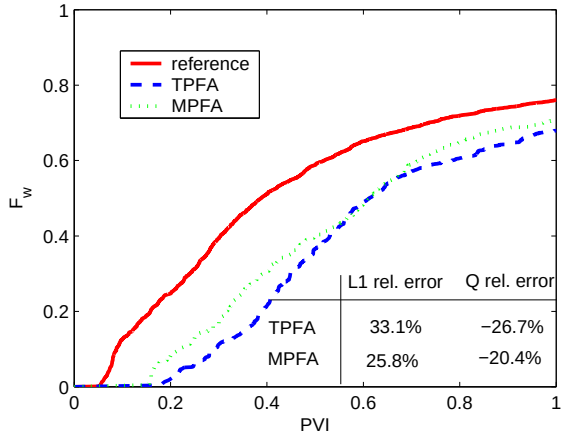
(b) $10 \times 5 \times 1$ (1152 nodes)



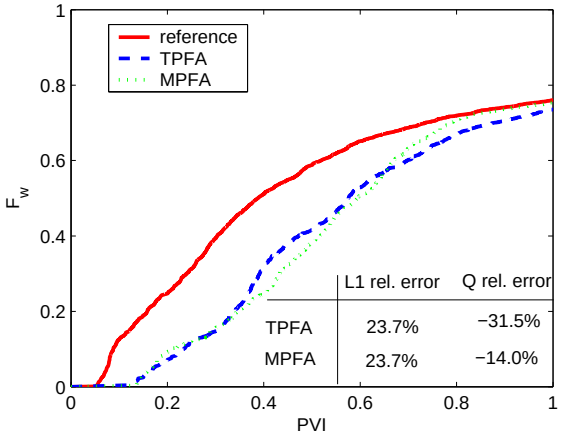
(c) $10 \times 5 \times 2$ (891 nodes)



(d) $10 \times 5 \times 2$ (1573 nodes)

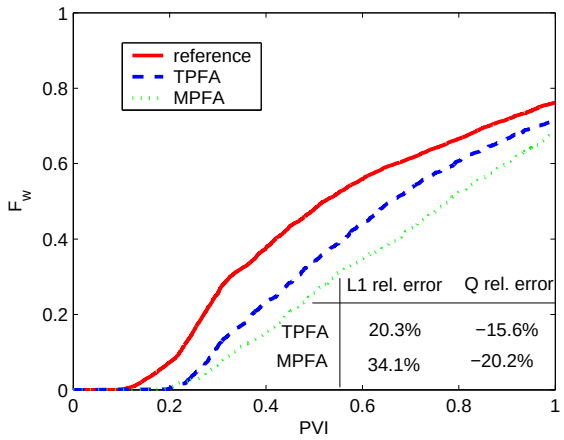


(e) $1 \times 1 \times 1$ (765 nodes)

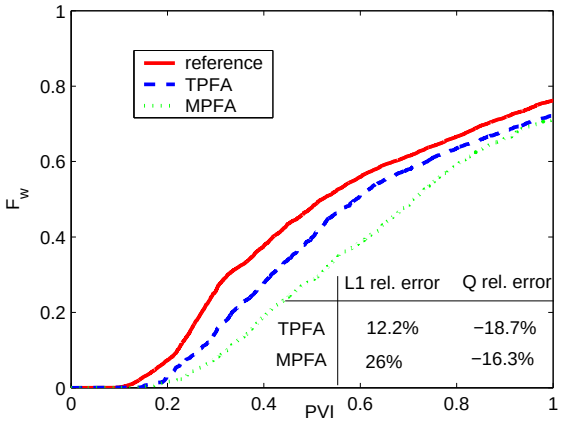


(f) $1 \times 1 \times 1$ (1381 nodes)

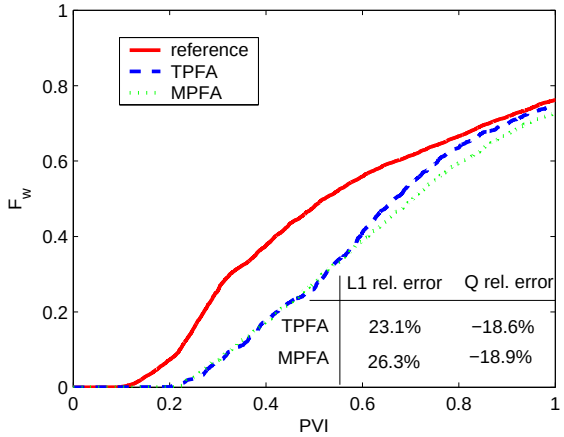
Figure 5.14: Comparison of T^*-TPFA and $\mathbf{k}^*$-MPFA upscalings for different uniformly upscaled coarse models (left-to-right flow)



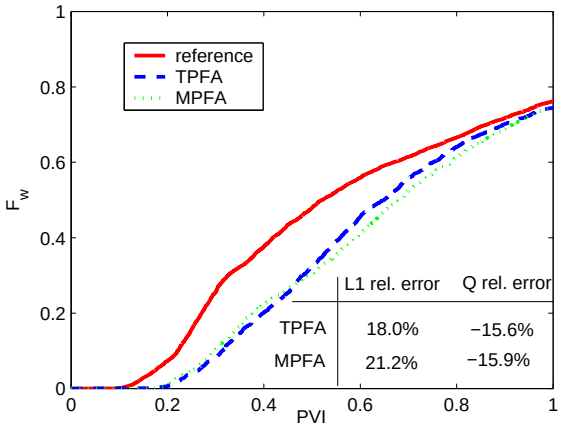
(a) $10 \times 5 \times 1$ (468 nodes)



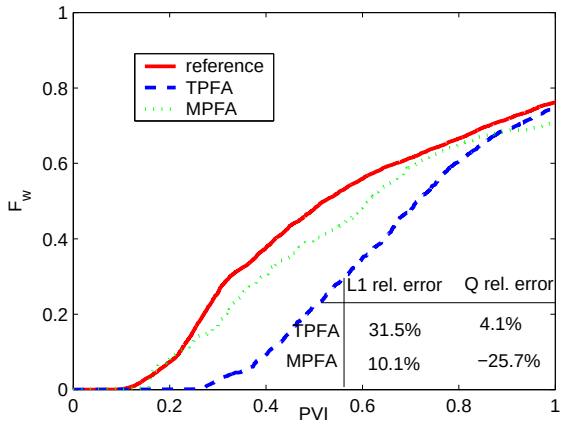
(b) $10 \times 5 \times 1$ (1152 nodes)



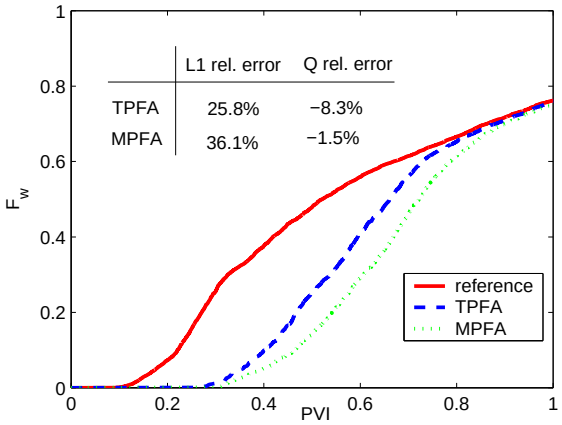
(c) $10 \times 5 \times 2$ (891 nodes)



(d) $10 \times 5 \times 2$ (1573 nodes)



(e) $1 \times 1 \times 1$ (765 nodes)



(f) $1 \times 1 \times 1$ (1381 nodes)

Figure 5.15: Comparison of T^*-TPFA and $\mathbf{k}^*$-MPFA upscalings for different uniformly upscaled coarse models (front-to-back flow)

to the reference solution, but rather the study of the relative accuracy when going from $\mathbf{k}^*$-MPFA to T^*-TPFA. The results show that the T^* upscaling usually performs better in terms of total $Q/\Delta P$ and the accuracy of the water cut curves.

For the left-to-right flow problem, the T^*-TPFA reduces the relative error in $Q/\Delta P$ from about 30% to less than 6% for all anisotropic grids (Figures 5.14(a)-(d)). For approximately isotropic grids (aspect ratio equal $1 \times 1 \times 1$), the same improvement is not observed. In particular, the results in Figure 5.14(f) show that the flow rate prediction significantly deteriorates using T^*-TPFA. This however occurs at a level of agreement with the reference where neither the T^*-TPFA or the $\mathbf{k}^*$-MPFA are particularly good. This may indicate that the grid is too coarse and/or the aspect ratio is inappropriate to capture the fine model flow characteristics with the local upscaling methods proposed.

For the front-to-back flow, none of the grids considered demonstrates a close agreement in terms of both $Q/\Delta P$ and the fractional flow curve. Results for the anisotropic grids are shown in Figures 5.15(a)-(d) and demonstrate about the same level of performance of T^*-TPFA and $\mathbf{k}^*$-MPFA in terms of $Q/\Delta P$. The fractional flow curve is nevertheless always improved using T^*-TPFA, in particular for the $10 \times 5 \times 1$ aspect ratio as shown in Figures 5.15(a) and (b). For the isotropic grid case shown in Figures 5.15(e) and (f), we cannot draw clear conclusions as to which method performs better.

The results presented above indicate that the T^*-TPFA method often performed better than the $\mathbf{k}^*$-MPFA for anisotropic grids. For isotropic grids, $\mathbf{k}^*$-MPFA generally performed better. The TPFA approach nonetheless demonstrated a high level of robustness to general anisotropy in all the cases of grid adaptation considered in this work. This represent a significant advantage over the less robust MPFA discretization considered here.

5.2.3 Remark on permeability resampling

In Section 4.2 detailing the T^* calculation, we described the rotation needed in order to map the permeability field onto an extended Cartesian region aligned with the direction of the transmissibility being calculated. This operation introduces a grid effect that may distort the geometry of the underlying permeability distribution (see Figure 4.6). The dimensions of the Cartesian cells in the rotated region can be the same or a fraction of the dimensions of the cell in the original grid. This defines what we call the resampling frequency. A resampling frequency equal to 2 therefore means that the cells in the rotated grid are half the size of those in the original grid.

We now illustrate the benefit of using a resampling frequency greater than 1. The reservoir model considered here is an oriented system of dimensions $L_x = 1$, $L_y = 1$ and $L_z = 0.5$ and contains $100 \times\ 100 \times 50$ Cartesian cells. The permeability distribution shown in Figure 5.16 is oriented relative to the coordinate system. We consider a left-to-right flow pattern where pressures are fixed at $x = 0$ and $x = L_x$. Two grids are considered in the comparison: (a) a uniform unstructured grid and (b) an adapted grid obtained using fine scale flow rate information.

Uniform grid

The grid is composed of 5120 nodes and each building cube has an aspect ratio of $5 \times 5 \times 2$ and represents a cubic region of the fine model containing $7 \times 7 \times 3$ Cartesian cells. The impact of the grid rotation on the quality of the upscaling can be observed in Figure 5.17 where the transmissibility upscaling was performed both with and without permeability resampling. In the case with resampling, the resampling frequency was chosen to be two ($s = 2$). Results show that the relative error of $Q/\Delta P$ is reduced from 10.5% to 0.3%. The

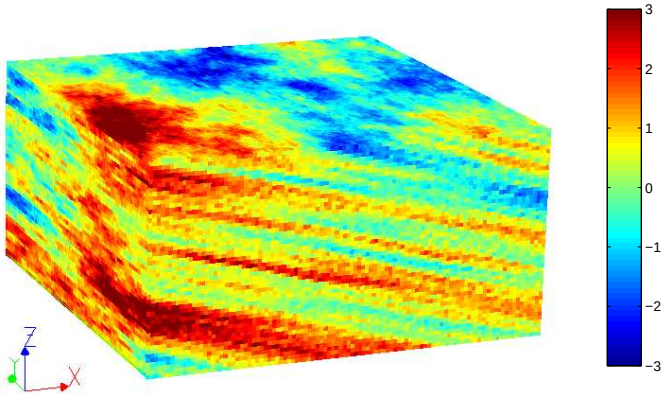
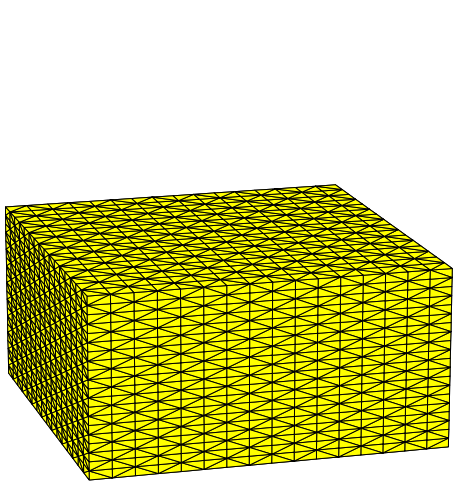


Figure 5.16: Permeability model of the oriented system ($\log$ scale)

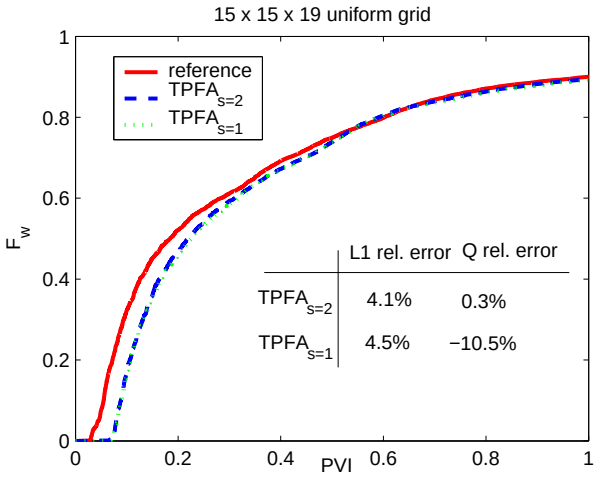
fractional flow curve is not affected by the resampling in that case.

Adapted grid

This grid, shown in Figure 5.18, was obtained by adapting the grid density to a mean flow rate map calculated on the fine model. It contains 1891 nodes. The flow results again show an appreciable improvement of the global $Q/\Delta P$ results from the resampling: the $Q/\Delta P$ relative error is reduced from 5.3% to only 0.5%, while the water cut curve is again essentially unchanged. Note that the errors observed using TPFA with resampling on the adapted grid are comparable to those on the coarse uniform grid. Yet, the number of nodes of the adapted grid is about one third of the number of nodes of the uniform grid, illustrating the benefit of using flow rate information in the grid adaptivity. The potential benefits of grid adaptivity will be further illustrated in the next chapter.

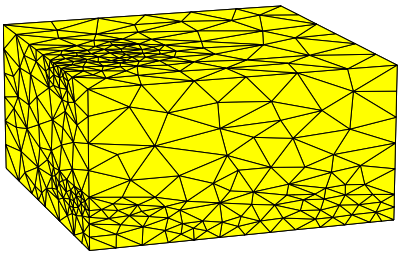


(a) Primal grid

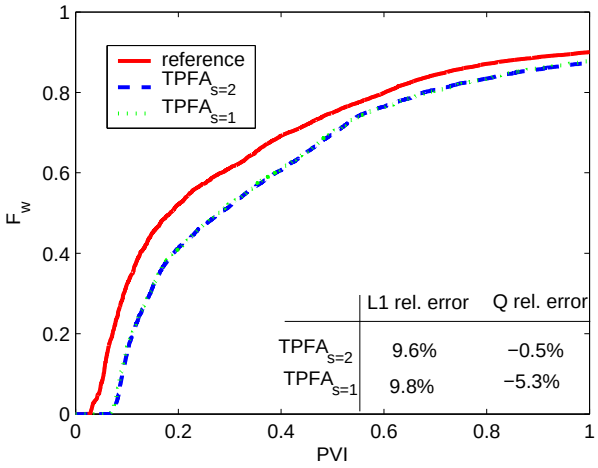


(b) Fractional flow comparison

Figure 5.17: Resampling of the permeability with a frequency greater than one provides a more accurate transmissibility upscaling (uniform grid, left-to-right flow)



(a) Primal grid



(b) Fractional flow comparison

Figure 5.18: Improvement of global $Q/\Delta P$ using a resampling with a frequency of 2 (adapted grid, left-to-right flow)

Chapter 6

Flow based gridding results

This chapter presents examples of flow-adapted grid generation. The tools developed and tested in previous chapters are used in a general methodology that aims at generating coarse reservoir models that preserve the essential flow characteristics of a fine Cartesian grid. The coarse grids generated are composed only of tetrahedra. We calculate the upscaled permeabilities or transmissibilities using the flow-based approach described in Chapter 4. Then, we test the models generated using the new streamline based unstructured simulator, assuming tracer flow, and using the upscaled permeabilities or transmissibilities. The examples presented illustrate the effects of gridding parameters for the flow adaptation such as flow information maps, grid aspect ratio and the parameterized grid resolution constraint.

The first section considers the selection of parameters for the grid adaptation. Next, we apply the overall methodology on a synthetic case. The example presented is a one million cell layered reservoir for which we propose an adapted grid that accurately captures the fine scale flow response for a particular flow pattern.

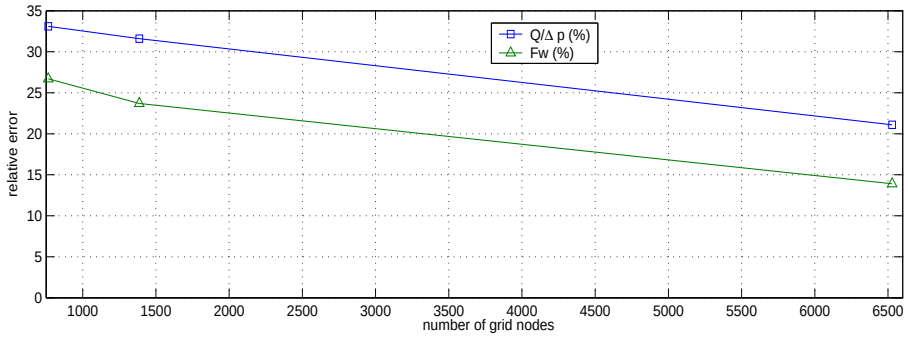
6.1 Parameter selection

We discuss here the strategy applied for selecting "optimized" parameters for the grid adaptation.

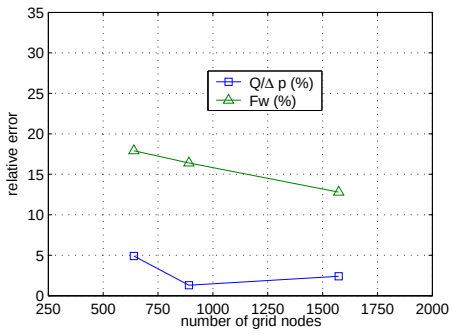
6.1.1 Aspect ratio

For the sake of reducing the number of optimization parameters, the 'target' grid anisotropy is first determined on uniformly coarsened grids. For this purpose, tracer flow simulation is used on different sized models for each aspect ratio. The error introduced by the coarsening is measured for different flow problems. The optimum aspect ratio is defined as the aspect ratio that provides the most accurate coarse models (relative to the reference solution). Then, when we consider unstructured grid adaptation, we preserve this optimal aspect ratio of the tetrahedral cells.

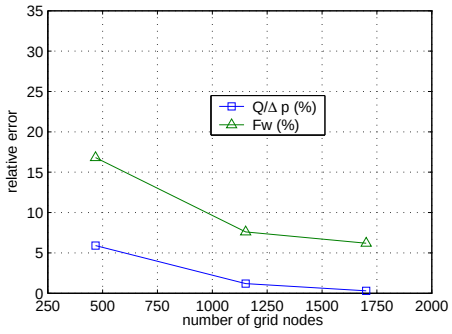
The optimum target aspect ratio is selected by first considering several possible aspect ratios and performing flow simulations on uniform grids of different sizes. Then the observed errors are plotted for each aspect ratio as a function of the grid size and the optimum value is selected. Figure 6.1 shows an example of the selection of the optimum aspect ratio. The model considered is introduced in Section 6.2. The problem used for the comparative convergence study is a left-to-right flow. It appears that the $10 \times 5 \times 1$ ratio is the best of the three possible ratios considered in this case. Specifically, using 1701 nodes, the error in $Q/\Delta P$ is 0.3% and the error in F_w is 6.2%. Note that the error does not always decrease monotonically (*e.g.*, Figure 6.1(b)); this is not unusual for upscaled models.



(a) aspect ratio is $1 \times 1 \times 1$



(b) aspect ratio is $10 \times 5 \times 2$



(c) aspect ratio is $10 \times 5 \times 1$

Figure 6.1: Comparison of flow response mismatch between uniform coarse tetrahedral grid and reference Cartesian grid for different aspect ratios (left-to-right flow)

6.1.2 Proportion of refined cells

The resolution constraint is assumed to have the simplified form discussed in Section 4.4. It will be used in practice to transform information map values such as flow rates into an indicator of whether a point insertion is needed for a given tetrahedron. The resolution constraint is defined by the information map and the four parameters P_a, P_b, s_a and s_b. For flow rate adaptation, our choice was to use values of P_a and P_b corresponding to the 25^{th} and 75^{th} percentiles. In some cases, we also considered the P_{75} and P_{100} in order to tailor the grid adaptation toward capturing very high flow regions.

For breakthrough-time adaptation, the shape of the resolution constraint function depends on which time interval of the map we wish to consider for the "fine" grid. If we are interested specifically in characterizing the breakthrough time, the shape proposed earlier for the resolution constraint is appropriate. We then use $P_a = P_0$ and $P_b = P_5$.

The results presented in Sections 6.2 use these conjectures without any sensitivity study. Further investigations are required to determine more systematically optimal P values. These values are, however, likely to be case dependent.

6.1.3 Grid iteration

In order to perform the adaptation, an initial tesselation must be provided. It is constructed using three parameters s_i° ($i \in \{1, 2, 3\}$) that define the size of the elements of dimension i (s_1° for edges, s_2° for triangles...). We now enumerate the steps involved in the construction of the adapted reservoir grid from the fine Cartesian information map:

1. Construct a reservoir model bounded (constrained) by intersecting surfaces (in transformed space)

2. Discretize the intersection lines with an initial grid size s_1°

3. Perform a triangulation of all the constraint surfaces, possibly with a target uniform size s_2°

4. Perform a tetrahedrization of the model, possibly with a target uniform size s_3°

5. Create a resolution constraint that assigns the relation between the value of the information map in the vicinity of a tetrahedron and its target size (radius of its circumscribed sphere)

6. Enforce the resolution constraint: identify all the tetrahedra violating the resolution constraints, then

   (a) Insert new vertices, creating new tetrahedra

   (b) Evaluate the new tetrahedra quality, possibly adding them to the list of tetrahedra to be refined

   (c) Iterate

Note that the initial grid resolution (s_3°) should be greater or equal to the largest grid size prescribed by the resolution constraint ($s_3^\circ \geq s_{\text{max}}$).

6.1.4 Example

We first give some illustration of the capabilities of the gridding methodology. The examples presented here were generated using the StanLab plug-in. We consider a simple geometry reservoir intersected by a single slanted fault. In addition to the structural constraints (Figure 6.2), we wish to adapt the grid density to the property distribution (Figure 6.3). A

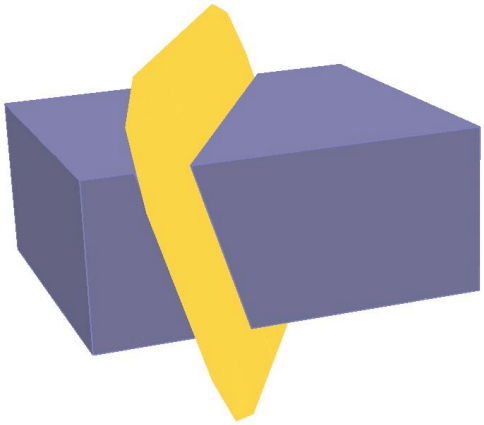


Figure 6.2: Three dimensional structural model of a faulted reservoir

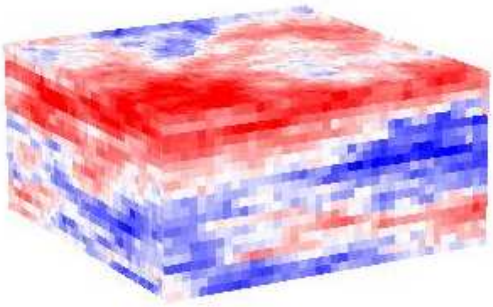


Figure 6.3: Three-dimensional model of permeability ($40 \times 40 \times 20$ cells)

flow map is obtained (Figure 6.4) from the permeability distribution by averaging the velocities of three flow problems (flow driven by pressure drops in each coordinate direction). A correspondence between flow rate and the desired grid resolution range is established (Figure 6.5). The corresponding resolution constraint is added to the unstructured grid model and a 3D tetrahedral grid is obtained. Finally, the upscaled permeabilities are calculated on the dual grid as explained in Section 4.1. The norms of the resulting permeability tensors are shown in Figures 6.6 and 6.7 for different views of the reservoir.

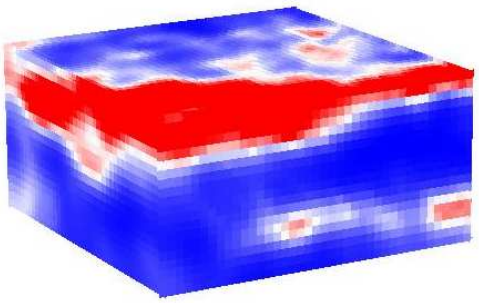


Figure 6.4: Local average flow velocities (average of three flow solutions)

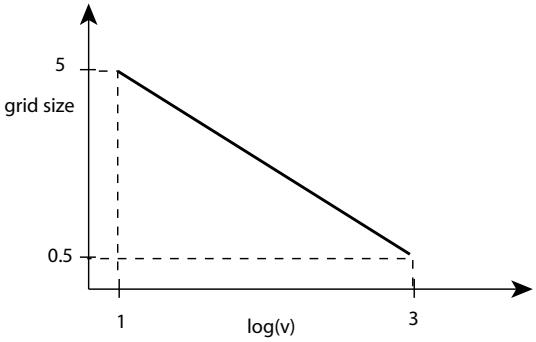


Figure 6.5: Prescribed correspondence between calculated average flow velocity and grid size

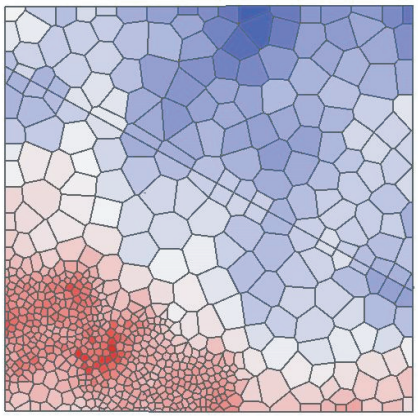


Figure 6.6: Top view of the upscaled model (dual grid) of the faulted reservoir ($|\mathbf{k}^*|$)

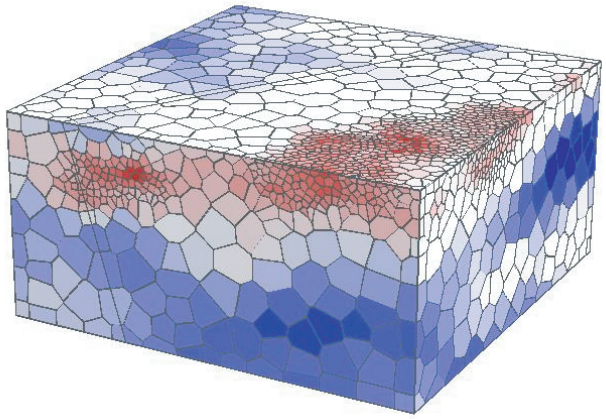


Figure 6.7: Side view of the upscaled model (dual grid) of the faulted reservoir ($|\mathbf{k}^*|$)

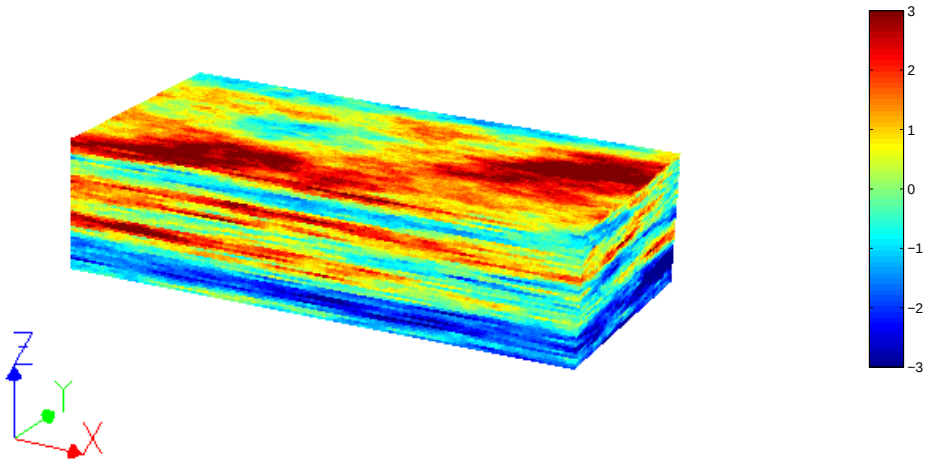


Figure 6.8: Layered reservoir, $\log k$

6.2 Layered reservoir

The reservoir considered in this application is a $200 \times 100 \times 50$ cell model populated with a log-normally distributed permeability field ($\sigma_{\log} = 1$, $\mu_{\log} = 0$). Here, $L_x = 1$, $L_y = 0.5$ and $L_z = 0.25$. Permeability is characterized by a spherical variogram with principal directions aligned with the coordinates axes and with the following dimensionless ranges: $\gamma_x = 1$, $\gamma_y = 0.75$ and $\gamma_z = 0.2$ (with γ_x nondimensioned by L_x, γ_y by L_y, and γ_z by L_z). A view of the reservoir is provided in Figure 6.8. In this section, we investigate grid generation using single phase flow rate and breakthrough time maps obtained from tracer streamline simulations.

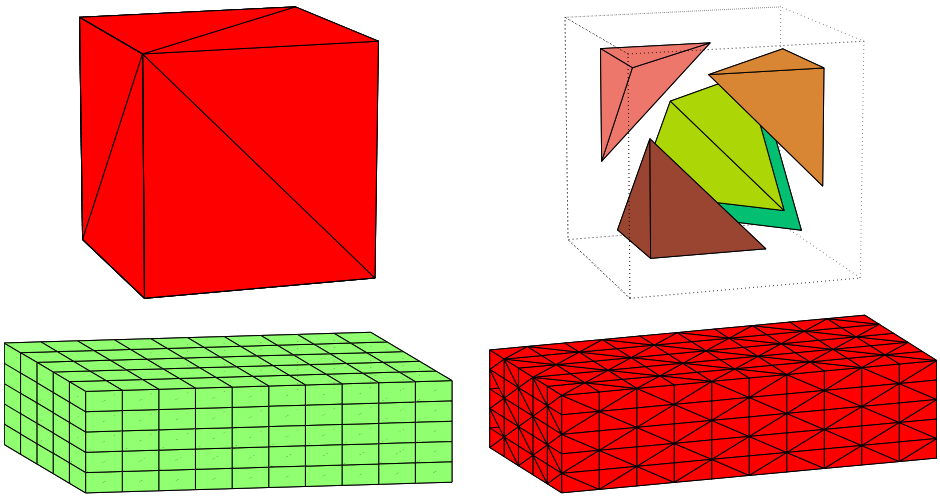


Figure 6.9: The uniform unstructured grids are constructed by splitting each Cartesian cell into five tetrahedra

6.2.1 Optimum target aspect ratio

As it is often the case for reservoir property models, the permeability distribution considered here exhibits a large degree of statistical anisotropy. A greater correlation of the permeability exists in the horizontal directions than in the vertical direction. We therefore anticipate that the grid resolution should be greater vertically than areally in the coarse reservoir representation. The first step of the gridding process is to determine an optimum cell aspect ratio for the unstructured grid.

We consider here three different aspect ratios for the uniform coarse grids. The coarse grid is unstructured, but obtained from a Cartesian coarse grid where each cell is split into five tetrahedra (see Figure 6.9). In view of the permeability correlations, we consider the three aspect ratios: $1 \times 1 \times 1$, $10 \times 5 \times 2$ and $10 \times 5 \times 1$. For each aspect ratio, several uniform grids of different sizes are considered. For each, tracer flow simulation is performed. Pressures are fixed on the $x = 0$ and $x = L_x$ faces ('left to right' flow). Each flow response is compared to the reference solution in terms of the relative error of global

Left to right flow			
Aspect ratio	Number of nodes	Error in $Q/\Delta P(\%)$	Error in F_w (%)
$1 \times 1 \times 1$	$17 \times 9 \times 5 = 765$	-26.7	33.1
	$21 \times 11 \times 6 = 1386$	-31.6	23.7
	$31 \times 16 \times 11 = 6528$	-21.1	13.9
$10 \times 5 \times 2$	$8 \times 8 \times 10 = 640$	-4.9	17.9
	$9 \times 9 \times 11 = 891$	-1.3	16.4
	$11 \times 11 \times 13 = 1573$	-2.4	12.8
$10 \times 5 \times 1$	$6 \times 6 \times 13 = 468$	-5.9	16.8
	$8 \times 8 \times 18 = 1152$	-1.2	7.6
	$9 \times 9 \times 21 = 1701$	-0.3	6.2
	$11 \times 11 \times 26 = 3146$	0.8	3.5

Table 6.1: Relative errors between fractional flow curves (L_1-norm) and $Q/\Delta P$ for the layered system using uniformly coarsened tetrahedral grids of different aspect ratios ('left-to-right' flow)

$Q/\Delta P$ and the L_1 norm of the error in the fractional flow curve, taken between 0 and 1 PVI.

The results are summarized in Table 6.1. We considered grids with a number of nodes varying between 468 and 6528. For a grid aspect ratio of $1 \times 1 \times 1$, the errors measured are quite large for the coarser grids (about 30% error in $Q/\Delta P$ and F_w for about 1000 nodes). When the number of nodes increases, the errors slowly reduce (21% error for $Q/\Delta P$ and 14% error for F_w for a 6528 node grid). This aspect ratio introduces a non-optimal node distribution, as a greater density of nodes should be used in the vertical direction than in the horizontal directions.

Improved overall results can be observed for the $10 \times 5 \times 2$ aspect ratio. The error in $Q/\Delta P$ is reduced to a few percent for 1573 nodes, while the error in F_w is 13%. For an aspect ratio of $10 \times 5 \times 1$, the results are improved over the results obtained with similar grid sizes but different aspect ratios. The error in $Q/\Delta P$ drops below 1% with 1701 nodes while the error in F_w is about 6%. The $10 \times 5 \times 1$ aspect ratio also exhibited the best rate

of convergence toward the reference results in terms of both global $Q/\Delta P$ and water cut. From now on for this case, we fix the aspect ratio of all grids generated to $10 \times 5 \times 1$. This can only be achieved in an approximate sense as explained in the following section.

6.2.2 Adaptation

Target aspect ratio

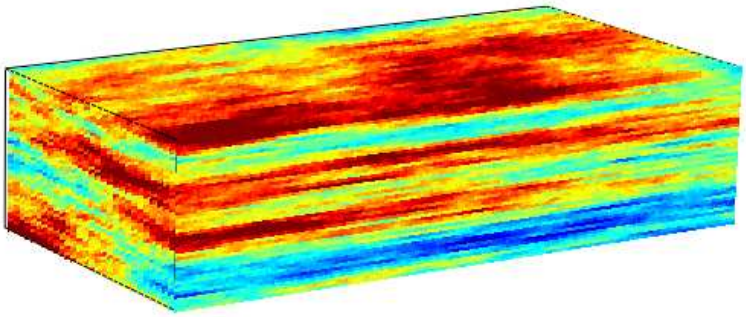
In this work, the 3D grid adaptation is performed by iterative insertion of points and local updating of the grid topology. The insertion criteria are provided by a grid resolution map and the updating is done in a way that forces the new grid to be strongly Delaunay[1] (in Euclidean space). In order to control the cell aspect ratio, the grid adaptation is performed in a transformed space (along the lines described in Section 6.1). In our example, where an aspect ratio of $10 \times 5 \times 1$ is used, the x-coordinate is left unchanged, the y-coordinate is stretched by a factor 2, and the z-coordinate is stretched by a factor of 10 in the transformed space.

6.2.3 Flow-rate adaptation

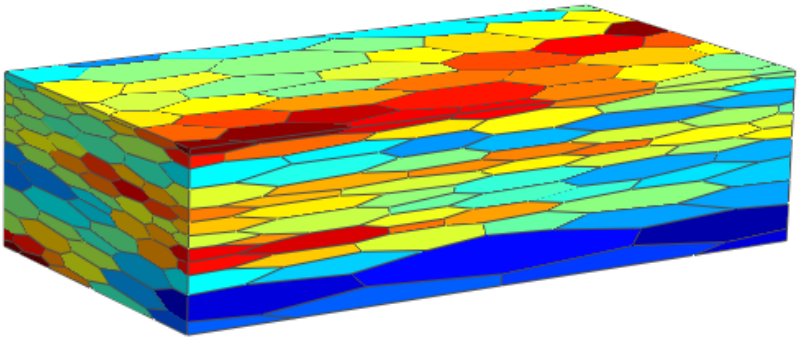
Resolution constraint

We first adapt the grid generation of the layered model using a mean flow-rate map obtained from single phase flow calculations. At each point in the Cartesian model, the value of the map is equal to the average norm of the local velocity vectors obtained by the three flow solutions. Boundary conditions were applied in each fine scale simulation so that the global pressure drop $(P_1^i - P_0^i)/L^i$ is constant and equal to one for each simulation.

[1] this means the Delaunay "empty sphere" property is satisfied for all nodes, including those with geometrical constraints



(a) $\log k$ on the Cartesian grid



(b) $\log |\mathbf{k}^*|$ on the dual grid

Figure 6.10: Fine and upscaled permeability fields

Because we wish to assign a smaller grid size to high flow regions, the grid resolution constraint is a piecewise-linear decreasing function of $\log \overline{V}$, where $\overline{V}$ is the mean velocity norm obtained from the three flow problems. The parameters a and b are chosen to be respectively the P_{75} and P_{25} values. We are considering an upscaling factor of about 1000 (*i.e.*, 1000 times fewer nodes in the coarse model than in the fine), therefore coarse model sizes may vary from a few hundred to a few thousand nodes.

Small-to-large ratio

In the present study, the largest-to-smallest grid size ratio (s_b/s_a) is fixed and tracer flow simulations are performed for models of different sizes (number of nodes). The error in the coarse grid simulations is then measured. The rate of reduction of the error as a function of the number of cells is then assessed and an optimum s_a/s_b is determined.

Optimum parameters

Table 6.2 shows the flow simulation results obtained by varying the number of grid cells N for a given small-to-large ratio s_a/s_b. This is done for three different values of s_a/s_b: $1/2$, $1/3$ and $1/4$. The ratio $s_a/s_b = 1/3$ gives the fastest rate of convergence of the errors. In particular, with 1394 nodes, the flow-rate adapted grid leads to an error in $Q/\Delta P$ of only 0.3% and an error in F_w of 5.2%. The value chosen for s_a/s_b and N are respectively $s_a/s_b = 1/3$ and $N = 1400$

6.2.4 Breakthrough-time adaptation

As discussed for 2D adapted gridding in Chapter 2, the grid can be adapted to attempt to capture the breakthrough time of a particular flow problem. Using the arrival time of each

Left to right flow			
s_a/s_b	Number of nodes	Error in $Q/\Delta P$ (%)	Error in F_w (%)
$1/2$	449 nodes	-0.8	13.4
	775 nodes	-0.7	9.4
	1580 nodes	-1.7	5.2
$1/3$	428 nodes	-4.4	14.7
	1394 nodes	0.3	5.2
$1/4$	405 nodes	-2.3	15
	1326 nodes	0.2	6.8

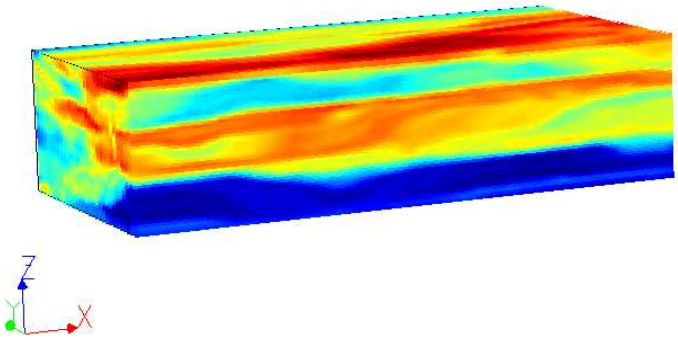
Table 6.2: Relative errors between fractional flow curves (L_1-norm) and $Q/\Delta P$ for the layered system using flow-rate adapted tetrahedral grids with different 'small-to-large' grid size ratios ('left-to-right' flow)

Left to right flow			
s_a/s_b	Number of nodes	Error in $Q/\Delta P$ (%)	Error in F_w (%)
$1/2$	832 nodes	1.1	12.4
	1907 nodes	5	8.2
$1/3$	511 nodes	0.7	15.7
	1140 nodes	12.9	10.3

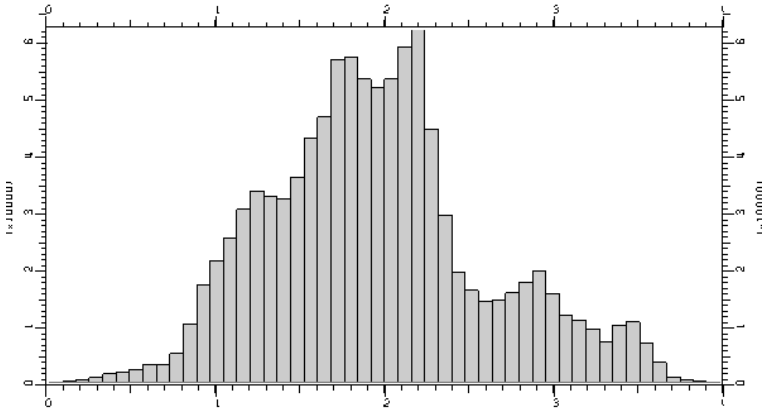
Table 6.3: Relative errors between fractional flow curves (L_1-norm) and $Q/\Delta P$ for the layered system using breakthrough-time adapted tetrahedral grids with different 'small-to-large' grid size ratios ('left-to-right' flow)

streamline and assigning its value to all traversed grid cells, a breakthrough time map can be generated and used as a grid adaptation property map. For the 'left-to-right' problem, the breakthrough time map is shown in Figure 6.11.

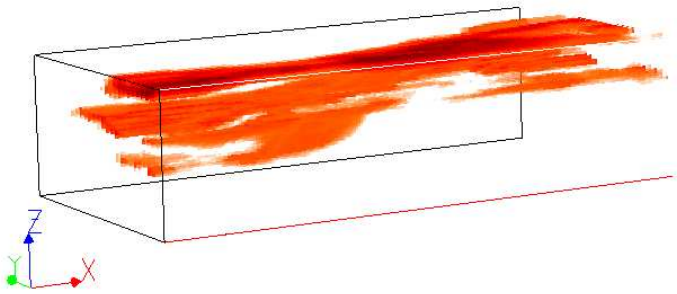
The grid construction is then adapted to capture regions of the reservoir swept by water at early time, corresponding to small values of the breakthrough-time map (see Figure 6.11). Table 6.3 summarizes the flow results obtained for adapted grids using s_a/s_b ratios of $1/2$ and $1/3$. The target grid aspect ratio was fixed to $10 \times 5 \times 1$ as previously. Values for the error in $Q/\Delta P$ are acceptable but deteriorate as we increase the number of nodes. The error in F_w is still significant, but the breakthrough time is captured accurately (see, for example, Figure 6.12, in which breakthrough time is perfectly captured by a 511



(a) log of the breakthrough time map



(b) Histogram of the log of the breakthrough time map



(c) First quartile of the log of the breakthrough map

Figure 6.11: Small values of the breakthrough-time map can be used to highlight regions of the reservoir swept by water at early time

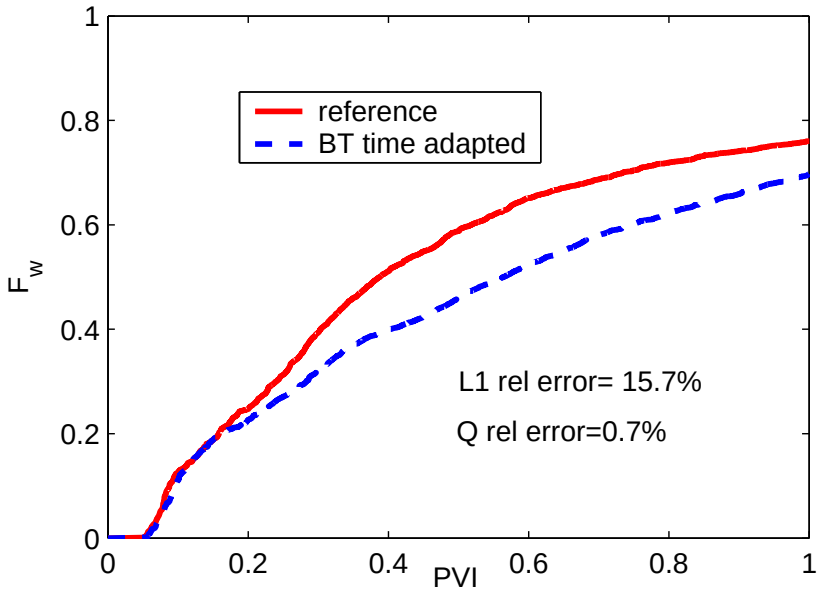


Figure 6.12: Using breakthrough time map can significantly improve the breakthrough time in the upscaled model

node adapted grid although there is subsequent error in F_w). In addition to an accurate breakthrough time, the $Q/\Delta P$ is also captured accurately (less than 1% error).

6.2.5 Breakthrough-time and flow-rate adaptation

Finally, use of both the flow rate map and the breakthrough-time map may provide better results than those using only a single map. For a fixed number of grid nodes, this amounts to shifting vertices from high flow rate regions to low breakthrough time regions. Figure 6.13 shows that when using breakthrough time information in addition to flow rate, the match of the fractional flow curve can be improved. The agreement in $Q/\Delta P$ degrades slightly, though it is still quite accurate. Further investigations are required to determine how these two flow information maps can be combined in an optimal manner.

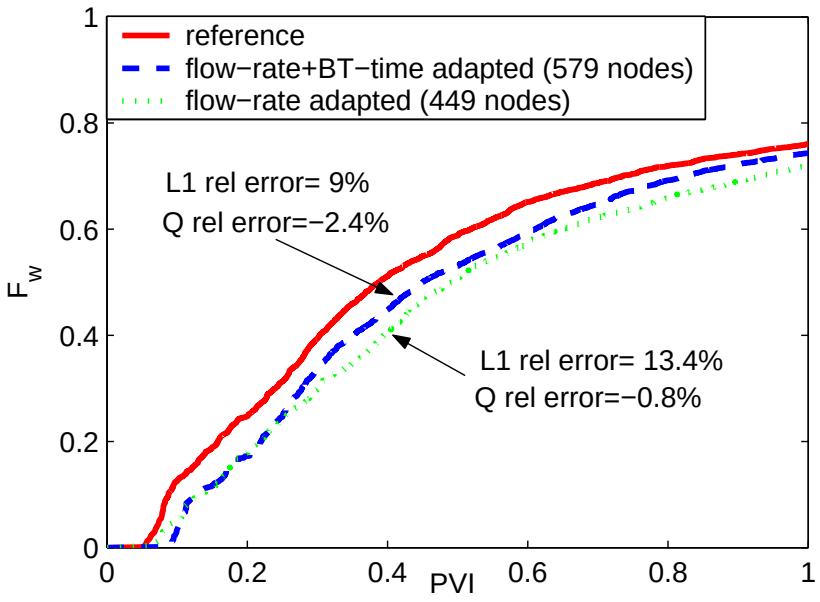
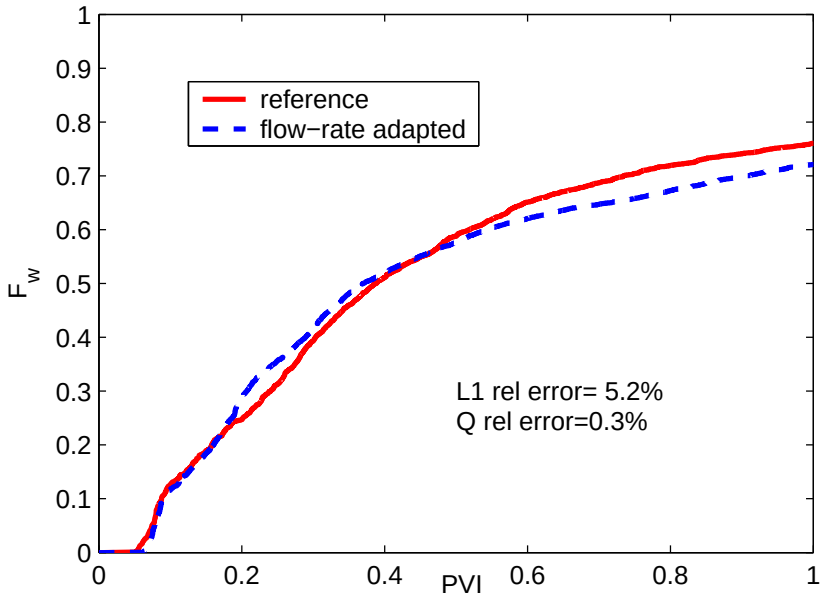


Figure 6.13: Using a resolution constraint that combines flow-rate and breakthrough time may reduce the F_w error (compared to using flow rate only)

6.2.6 Discussion

The flow adaptation was performed in three steps: (1) determine the optimum target aspect ratio from uniformly coarsened grids, (2) determine the optimum large-to-small cell size ratio, (3) select the grid size (number of cells) to achieve the desired level of accuracy (based on the flow diagnostics).

Both flow rate and breakthrough time adaptation were tested. Flow rate adaptation allowed us to improve the flow results for the left-to-right flow problem when compared to a uniform grid with the same number of nodes. The breakthrough time, however, did not exhibit systematic improvements. The left-to-right tracer problem proved to be captured accurately with only 1400 nodes using a flow-rate adapted grid (less than 1% error in the fractional flow curve and within 5% of the total flow rate) as shown in Figure 6.14. Such a match could not be achieved with comparable grid sizes using uniform grids. Figure 6.15

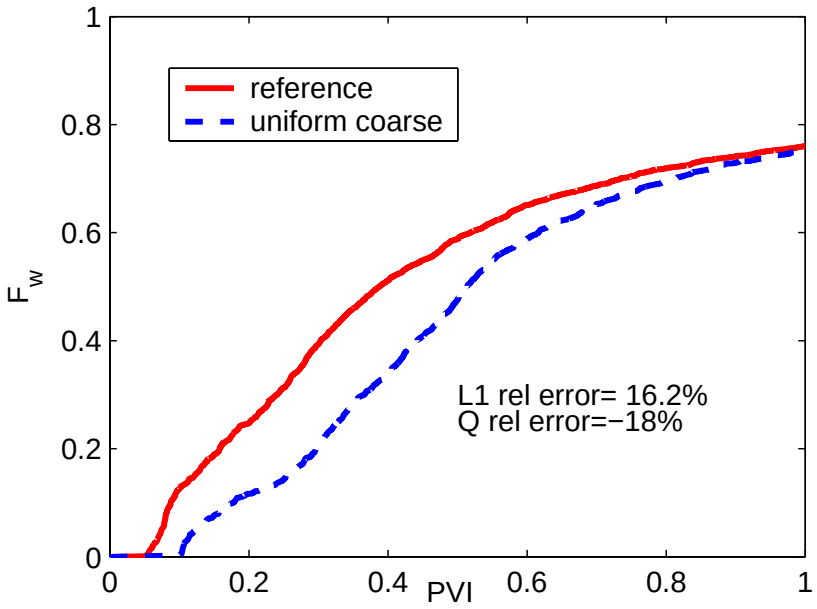


(a) Left-to-right

Figure 6.14: Best flow rate adapted grid (1394 nodes); aspect ratio is $10 \times 5 \times 1$ and $s_a/s_b = 3$

shows a uniformly coarsened grid obtained after "guessing" a grid aspect ratio of $2 \times 2 \times 1$. Similar results for a $1 \times 1 \times 1$ aspect ratio are shown in Figure 6.16.

Finally, we consider a different flow problem using the grid established for the left to right flow scenario. For flow from front-to-back, we obtain the results shown in Figure 6.17 for (a) a uniform grid and (b) the flow-adapted grid determined for left-to-right flow. The uniform grid provides an accurate estimate of $Q/\Delta P$, though there is a noticeable error in F_w. The flow rate adapted grid, by contrast, provides accurate F_w but shows substantial error in $Q/\Delta P$. This illustrates that the optimal grid for one flow scenario may not be optimal for other flow scenarios. We note that, for this problem, errors for front-to-back flow were generally higher than those for left-to-right flow. This was found to be the case using both $\mathbf{k}^*$-MPFA and T^*-TPFA. It may also be possible that the large amount



(a) Left-to-right

Figure 6.15: Flow results for a uniformly coarsened grid with 2541 nodes and an "intuitive" aspect ratio

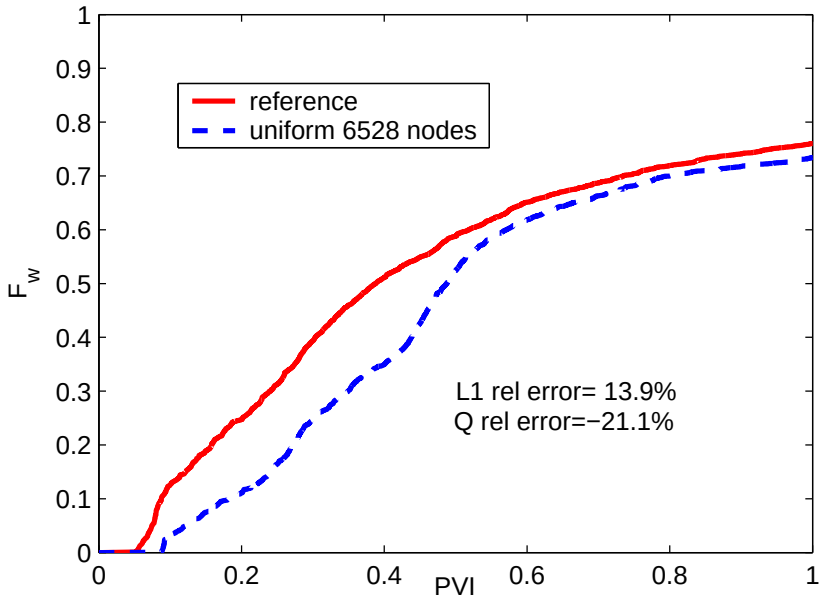
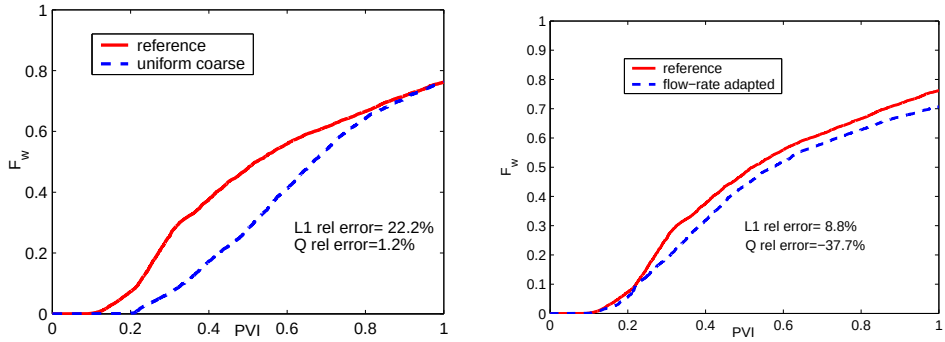


Figure 6.16: Flow results for a uniformly coarsened grid with 6528 nodes and an aspect ratio of $1 \times 1 \times 1$ (left-to-right flow)



(a) uniformly coarsened grid with 2541 nodes

(b) Grid selected based on the left-to-right match

Figure 6.17: Front-to-back flow problem proved not to be captured by grids selected using the left-to-right flow diagnostic

of upscaling we considered (a factor of 1000) is only appropriate for the left-to-right case. A significantly greater number of nodes may be needed to capture the front-to-back flow problem accurately.

Chapter 7

Conclusions and future directions

This work proposed a methodology for constructing accurate coarse reservoir models from detailed geocellular representations. The solutions developed allow for the resolution of complex reservoir geometries by optimizing the match between the coarse and fine model flow responses for a collection of flow problems. To achieve this, three key technologies were integrated into a general framework: flexible unstructured gridding, flow-based upscaling and streamline simulation. New developments in each of these domains have been accomplished and their accuracy and robustness have been demonstrated. The principal technical contributions of this work include: (a) a novel extension of flow based upscaling techniques for permeability tensors and transmissibilities on unstructured grids, (b) a streamline simulator based on control volume solutions on unstructured grids, and (c) a methodology for generating grids based on flow information.

7.1 Summary and conclusions

Unstructured grids are naturally suited for capturing complex geometries and resolving specific regions of the reservoir. Recent developments in the area of constrained grid generation have introduced new capabilities to coarse reservoir modeling as general geometrical constraints can now be resolved in a more efficient manner than via the use of structured corner point geometry grids. The development of streamline simulation techniques now broadens the field of applications for unstructured grids. This may in particular open the way for extensions of existing reservoir characterization techniques based on streamline simulation.

The principal challenge we encountered in developing the unstructured streamline simulator was the accurate reconstruction of the velocity field from the fluxes computed from the CVFE solution. The method proposed entails a local velocity postprocessing that involves the resolution of a system of constraints that guarantees continuity and consistency with the numerically obtained fluxes. This required further discretization of the unknown velocity field in order to obtain a system of minimum size that allowed us to enforce all the constraints. This solution introduces a tetrahedra sub-grid for the tracing. The postprocessing and the sub-grid tracing lead to a considerable increase of computational time and data storage when compared to simulations performed on equivalent-size Cartesian models. In this work, no special investigations were directed toward reducing cost, although we believe that it can be reduced significantly by improving the algorithms of the subgrid construction and storage. In particular one should take advantage of the primal grid structure to access the subgrid topology, via a hierarchical grid data structure for instance, as explained in the next section.

New flow-based upscaling methods were developed in this work. The extended local

transmissibility upscaling proved to be very efficient at capturing subscale permeability heterogeneity and anisotropy. It was also shown that it is often more accurate and generally more robust than the multiple-point flux approximation coupled with extended local permeability upscaling. A significant benefit of the T^*-TPFA is that it bypasses the (multiple-point) transmissibility calculation, hence saving the computational effort that is required for a sophisticated flux-continuous scheme. As the T^*-TPFA does not rely on any particular topology of the primal grid, it may be applicable to even more general unstructured grids, that may for example combine hexahedral cells, prisms, tetrahedra and/or pyramids.

The flow-based gridding examples demonstrated the potential benefits of the use of flow information in the grid adaptation process. The grid adaptation was presented in a parameterized form using a resolution constraint. A parameter selection method and a step by step procedure were also presented. The synthetic reservoir model considered was upscaled by a factor of about 1000 and showed good flow agreement with the fine model. This close agreement was due to the combined use of flow based transmissibility upscaling and flow rate grid adaptation. Further study of the quantitative benefits of the adaptation as well as default values for the gridding parameters are needed. Also, the robustness of the grid adaptation must be further investigated when simulating more complex displacements.

Finally, most of the tools used in this work have been integrated into a software development platform combining reservoir simulation specific functionalities as well as state of the art geomodeling, gridding and visualization capabilities. This platform, based on a Gocad kernel, allows us to perform all the tasks comprised in our grid generation methodology in a single environment. Thus, structural modeling, reservoir property construction, fine scale flow simulations (tracer), adapted gridding, upscaling and assessment of diagnostics (via unstructured tracer simulations) can all be performed within the same software

environment.

7.2 Future directions

The key future research directions include the enhancement and further testing of the techniques developed in this work. The speed of the streamline simulation and upscaling routines needs to be improved in order to be able to consider systems with larger numbers of grid nodes. Specific suggestions for future development are as follows.

7.2.1 Gridding

In the flow based gridding examples, we observed the important impact of the aspect ratio of the grid cells on the accuracy of the flow response. We proposed a simplified mapping approach to include some geometrical anisotropy in the grids generated. However, a more general solution is needed. We suggest that the resolution constraint be a tensor field instead of a scalar value, and that the cell quality criteria be evaluated in the corresponding metric space. This will provide variable grid anisotropy in a very natural and general fashion.

This work introduced two flow information maps (flow rate and breakthrough time maps). These information maps are not the only useful sources of information for the grid adaptation. The distance of a cell to a fault or a well may also be of some interest. A combination of average flow information and problem-specific distance maps may improve the gridding strategy.

7.2.2 Streamline simulation

Our implementation of the streamline simulator can be improved in several ways. Unstructured streamline simulation introduces a sub-grid for the tracing. The topology construction and storage needs to be improved considerably for this to be a practical tool. Velocity interpolants are constructed locally on a control volume basis. The current sub-grid topology does not take advantage of this aspect. A substantial speed up may be obtained on the sub-grid construction if it is stored as a hierarchical structure. In that case the sub-grid topology would have two levels: one local to the control volume and one global, connecting the control volumes together (the latter topology is already embedded in the primal grid and therefore need not be recomputed).

The use of the two-point flux approximation considerably simplifies the definition of the flux on the faces of the dual cells. Indeed, the flux between two nodes only depends on the pressure at these two nodes and does not involve the pressures at the vertices of other tetrahedra sharing the edge connecting the two nodes. As a consequence, the flux continuity constraints imposed on the postprocessed velocity become much simpler. In addition, their number reduces from $6n_t$ to n_e, where n_t is the number of tetrahedra and n_e is the numbed of edges in the grid. For a uniform grid for instance, this represents a reduction of a factor 5. This reduction in the number of constraints may be used to reduce the number of unknowns characterizing the velocity field, therefore significantly speeding up the postprocessing and the tracing itself when the two-point flux approximation is used.

7.2.3 Upscaling, MPFA and TPFA

Further investigations are required to determine practical criteria that identify cases when the current MPFA fails for the solution of the discretized pressure equation. It will be

useful to determine relationships between the conditioning of the matrix and the grid and permeability field.

It may be possible to extend the T^*-TPFA upscaling into a local-global procedure (along the lines of Chen *et al.*, 2003). This will require the use of the global pressure obtained from the coarse scale solutions and iterations of the upscaled transmissibilities until a self consistent solution is achieved.

Nomenclature

Notation	Definition
Roman	
a_i	Length of the ith principal axis of an ellipsoid
$\mathcal{C}$	Barycenter of a face
dl	Element of length
ds	Element of surface
dV	Element of volume
$\mathcal{E}$	Edge
f	Flux (flow rate)
F_w	Producing water fraction (water-cut)
$\mathcal{F}$	Triangular face
F_ij	Face between the two dual cells i and j
$\mathbf{G}$	Pressure gradient
$\mathcal{G}$	Barycenter of a tetraheron
$\mathcal{H}$	Hexahedron

continued on next page

Notation	Definition
$\mathbf{k}$	Permeability tensor
$\mathbf{k}^*$	Upscaled permeability tensor
l	Dimensionless correlation length or length of augmented system
L	System length
$\mathcal{M}$	Midpoint of edge
N	Total number of grid nodes
$\hat{\mathbf{n}}$	Normal vector
n_t	Number of tetrahedra connected to a node
n_f	Number of triangular faces connected to a node
P	Pressure or percentile
p	Vertex of the primal grid
Q	Total producing rate
q	Flow rate
$\mathcal{Q}$	Quadrilateral face (face of a control volume)
r	Number of rings in the border region
S_w	Water saturation
s_a, s_b	Minimum and maximum values of the target grid size
T	Transmissibility coefficient
T^*	Upscaled transmissibility
$\hat{\mathbf{t}}$	Tangential vector

continued on next page

Notation	**Definition**
t	Time
$\tilde{t}$	Simulation time expressed in PVI
$\mathcal{T}$	Tetrahedron
$\mathbf{u}$	Darcy velocity
u	Spatial direction ($u \in x, y, z$)
$\mathbf{V}$	Particle velocity interpolant
$\mathbf{V}^{\circ}$	Natural velocity
$\mathbf{V}^{*}$	Recovered velocity
V	Volume
$\mathbf{x}$	Cartesian coordinate vector
x, y, z	Physical space coordinates
Greek	
γ	Dimensionless range
Λ	Fluid mobility
λ	Eigenvalue
μ	Fluid viscosity
$\mu_{\log}$	Mean of the log of a distribution
σ	Singular value
$\sigma_{\log}$	Standard deviation of the log of a distribution
ϕ	Rock porosity
τ	Time of flight

continued on next page

Notation	Definition
ξ, η, ν	Reference space coordinates
Operators	
$\langle \cdot \rangle$	Spatial average
$\overline{\cdot}$	Arithmetic mean
$\lvert \cdot \rvert$	Area of a surface
diag$(v_1, \ldots, v_n)$	Diagonal matrix A with $A(i,j) = v_i$
null$(\cdot)$	Null space of a matrix
range$(\cdot)$	Range (image space) of a matrix
rank$(\cdot)$	Rank of a matrix, dimension of the image space
$\epsilon -$ rank	Numerical rank, number of singular values greater than ϵ
Abbreviations	
CVFE	Control volume finite element
FE	Finite element
IMPES	Implicit pressure explicit saturation
MPFA	Multiple-point flux approximation
PVI	Pore volume injected
TPFA	Two-point flux approximation

Bibliography

Aavatsmark, I., Barkve, T., Boe, O., and Mannseth, T. (1996). Discretization on non-orthogonal curvilinear grids for inhomogeneous, anisotropic media. *Journal of Computational Physics*, **127**(1), pp 2–14.

Aavatsmark, I., Barkve, T., Boe, O., and Mannseth, T. (1998). Discretization on unstructured grids for inhomogeneous, anisotropic media. Part I: Methods. *SIAM Journal on Scientific Computing*, **19**, pp 1700–1716.

Batycky, R. P., Blunt, M. J., and Thiele, M. R.(1997). A 3D field-scale streamline-based reservoir simulator. *SPE Reservoir Engineering*, **12**(4), pp 246–254.

Borouchaki, H., George, P.-L., Hecht, F., Laug, P., and Saltel, E. (1997). Delaunay mesh generation governed by metric specifications. 1. Algorithms. *Finite Elements in Analysis and Design*, **25**(1-2), pp 61–83.

Breitenbach, E. A., Thurnau, D. H., and Van Poolen, H. K. (1969). The fluid flow simulation equations. *SPE Journal*, **2**(9), pp 155–169.

Cao, H. (2002). *General Purpose Research Simulator, Reference Manual*.

Cao, J. and Kitanidis, P. (1999). Adaptive-grid simulation of groundwater flow in heterogeneous aquifier. *Advances in Water Resources*, **22**(7), pp 681–696.

Cardwell, W. T. and Parson, R. L. (1945). Averaging permeability of heterogeneous oil sands. *Transactions of the American Institute of Mining Metallurgical and Petroleum Engineers*, **160**, pp 34–42.

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

Chen, Y., Durlofsky, L. J., Gerritsen, M., and Wen, X. H. (2003). A coupled local-global upscaling approach for simulating flow in highly heterogeneous formations. *Advances in Water Resources*, **26**, pp 1041–1060.

Coats, K. H. (1979). *Elements of Reservoir Simulation*. Lecture Notes, University of Texas, reprinted by Intercomp Resources Development and Engineering Inc., Houston, Texas.

Cordes, C. and Kinzelbach, W. (1992). Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear elements. *Water Resources Research*, **28**(11), pp 2903–2911.

Cordes, C. and Kinzelbach, W. (1994). Can we compute exact pathlines in 3D ground water flow models? In *Tenth International Conference on Computational Methods in Water Resources*, pp 225–232, Universität Heidelberg, Germany.

Deutsch, C. V. (1989). Calculating effective absolute permeability in sandstone/shale sequences. *SPE Formation Evaluation*, **3**(4), pp 349–348.

Durlofsky, L. J. (1991). Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water Resources Research*, **27**(5), pp 699–708.

Durlofsky, L. J. (2003). Upscaling of geocellular models for reservoir flow simulation: A review of recent progress. In 7^{th} *International forum on reservoir simulation*, Bühl/Baden-Baden, Germany.

Durlofsky, L. J., Behrens, R. A., Jones, R. C., and Bernath, A. (1996). Scale up of heterogeneous three dimensional reservoir descriptions. *SPE Journal*, **1**, pp 313–326.

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. *Advances in Water Resources*, **20**(5-6), pp 335–347.

Edwards, M. G. (2002). Unstructured, control-volume distributed, full-tensor finite-volume schemes with flow based grids. *Computational Geosciences*, **6**(3-4), pp 433–452.

Edwards, M. G. and Rogers, C. F. (1998). Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Computational Geosciences*, **2**, pp 259–290.

Edwards, M. G., Agut, R., and Aziz, K. (1998). Quasi K-orthogonal streamline grids: Gridding and discretization, paper SPE 49072 presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Lousiana, 27-30 September.

Eek-Jensen, C. F., Aavatsmark, I., and Boe, O. (1999). Upscaling on general quadrilaterals in 3D with application to field cases. In *EAGE 10th European Symposium on Improved Oil Recovery*, Brighton, United Kingdom.

Forsyth, P. A. (1990). A control-volume, finite-element method for local mesh refinement in thermal reservoir simulation. *SPE Reservoir Engineering*, **5**, pp 561–566.

Fung, L. S.-K. and Nghiem, L. X. (1990). A control-volume finite-element scheme using triangular elements for reservoir simulation. *CGM Report*, March, **90.03.R**.

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

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. The Johns Hopkins University Press, Baltimore, MD, third edition.

Gunasekera, D., Childs, P., Herring, J., and Cox, J. (1998). A multi-point flux discretization scheme for general polyhedral grid, paper SPE 48855 presented at the SPE 6th International Oil & Gas Conference and Exhibition, Beijing, China, 2-6 November.

Hale, D. (2002). Atomic meshes: from seismic imaging to reservoir simulation. In *8th European Conference on the Mathematics of Oil Recovery*, Freiberg, Germany. http://sepwww.stanford.edu/oldsep/hale/Hale083002AtomicMeshes.pdf.

He, C., Edwards, M. G., and Durlofsky, L. J. (2002). Numerical calculation of equivalent cell permeability tensors for general quadrilateral control volumes. *Computational Geosciences*, **6**(1), pp 29–47.

Heinemann, Z. E., Heinemann, G. F., and Tranta, B. M. (1998). Modeling heavily faulted reservoirs, paper SPE 48998 presented at the SPE Annual Technical Conference, New Orleans, Louisiana, 27-30 September.

Heinemann, Z. E., Brand, C. W., and Munka, M. (1991). Modeling reservoir geometry with irregular grids. *SPE Reservoir Engineering*, **6**, pp 225–232.

Hughes, T. J. R., Engel, G., Mazzei, L., and Larson, M. G. (2000). The continuous Galerkin method is locally conservative. *Journal of Computational Physics*, **163**(2), pp 467–488.

Jenny, P., Wolfsteiner, C., Lee, S. H., and Durlofsky, L. J. (2002). Modeling flow in geometrically complex reservoirs using hexahedral multiblock grids. *SPE Journal*, **7**, pp 149–157.

Johnson, R. A. and Wichern, D. W. (1988). *Applied multivariate statistical analysis*. Prentice-Hall, Engelwood Cliffs, New Jersey.

Journel, A. G., Deutch, C. V., and Desbarats, A. J. (1986). Power averaging for block effective permeability, paper SPE 15128 presented at the SPE California Regional Meeting, Oakland, California, 2-4 April.

Lee, S. H., Durlofsky, L. J., Lough, M. F., and Chen, W. H. (1998). Finite difference simulation of geologically complex reservoirs with tensor permeabilities. *SPE Reservoir Evaluation & Engineering*, **1**, pp 567–574.

Lee, S. H., Tchelepi, H. A., Jenny, P., and DeChant, L. J. (2002). Implementation of a flux-continuous finite-difference method for stratigraphic, hexahedron grids. *SPE Journal*, **7**, pp 267–277.

Lepage, F. (2002). Triangle and tetrahedral meshes for geological models. In *Proceedings of the IAMG 2002 Annual Conference*, Berlin, Germany.

Lepage, F. (2003). *Generation de maillage adaptatifs structurés et hybrides pour la modelization de gisements*. Ph.D. thesis, Ecole National Superieure de Geology de Nancy.

Levy, B. (1999). *Topologie Algrithmique: Combinatoire et Plongement*. Thèse d'université, INPL. Prix SPECIF 2000.

Lévy, B. and Prévost, M. (2000). Structured and unstructured grids for reservoir simulation. In *Proceedings of the 20th Gocad Meeting*, Vandoeuvre, France.

Li, D. and Beckner, B. (2000). Optimal uplayering for scaleup of multimillion-cell geologic models, paper SPE 62927 presented at the Annual Technical Conference and Exhibition, Dallas, Texas, 1-4 October.

Mallet, J. L. (2002). *Geomodeling*. Oxford University Press, New York, New York.

Palagi, C. L. (1992). *Generation and application of Voronoï grids to model flow in heterogeneous reservoirs*. Ph.D. thesis, Stanford University.

Palagi, C. L. and Aziz, K. (1991). Use of Voronoï grids in resevoir simulation, paper SPE 22889 presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 6-9 October.

Pokrajac, D. and Lazic, R. (2002). An efficient algorithm for high accuracy particle tracking in finite elements. *Advances in Water Resources*, **25**(4), pp 353–369.

Prévost, M. (2000). *Streamline method for unstructured grids*. Master's thesis, Stanford University.

Prévost, M. (2002). Accurate reservoir modeling using unstructured grids and flow-based upscaling, presented at the SPE Annual Technical Conference and Exhibition (International Student Paper Contest), San Antonio, Texas, 29 September-2 October.

Prévost, M., Edwards, M. G., and Blunt, M. J. (2001). Streamline tracing on curvilinear structured and unstructured grids, paper SPE 66347 presented at the SPE Reservoir Simulation Symposium, Houston, Texas, 11-14 February.

Prévost, M., Edwards, M. G., and Blunt, M. J. (2002). Streamline tracing on curvilinear structured and unstructured grids. *SPE Journal*, **7**, pp 139–148.

Renard, P. and de Marsily, G. (1997). Calculating equivalent permeability: A review. *Advances in Water Resources*, **20**(5-6), pp 253–278.

Rubin, Y. and Gómez-Hernandéz, J. J. (1990). A stochastic approach to the problem of upscaling of conductivity in disordered media: Theory and unconditional numerical simulations. *Water Resources Research*, **24**(4), pp 397–409.

Settari, A. and Aziz, K. (1972). Use of irregular grids in reservoir simulation. *SPE Journal*, **2**(12), pp 103–114.

Shewchuk, J. R. (1997). *Delaunay refinement mesh generation*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Shewchuk, J. R. (1998). Tetrahedral mesh generation by Delaunay refinement mesh generation. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, pp 86–95, Minneapolis, Minnesota.

Shewchuk, J. R. (2000). Mesh generation for domains with small angles. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, pp 1–10, Hong-Kong.

Thiele, M. R. (1994). *Modeling multiphase flow in heterogeneous media using streamtubes*. Ph.D. thesis, Stanford University.

Tran, T. T. B. (1995). *Stochastic simulation of permeability fields and their scale-up for flow modeling*. Ph.D. thesis, Stanford University.

Trykozko, A., Zijl, W., and Bossavit, A. (2001). Nodal and mixed finite elements for the numerical homogenenization of 3D permeability. *Computational Geosciences*, **5**(5), pp 61–84.

Verma, S. K. (1996). *Flexible grids for reservoir simulation*. Ph.D. thesis, Stanford University.

Verma, S. K. and Aziz, K. (1997). A control volume scheme for flexible grids in reservoir simulation, paper SPE 37999 presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June.

Warren, J. E. and Price, H. H. (1961). Flow in heterogeneous porous media. *SPE Journal*, **1**, pp 153–169.

Wen, X. H. (1996). *Stochastic simulation of groundwater flow and mass transport in heterogeneous aquifers: Conditioning and problems of scale*. Ph.D. thesis, Polytechnic University of Valencia (Spain).

Wen, X. H. and Gómez-Hernández, J. J. (1996). Upscaling hydraulic conductivities in heterogeneous media: An Overview. *Journal of Hydrology*, **183**(1-2), pp ix–xxxii.

Wen, X. H., Durlofsky, L. J., Lee, S. H., and Edwards, M. G. (2000). Full tensor upscaling of geologically complex reservoir descriptions, paper SPE 62928 presented at the SPE Annual Technical Conference, Dallas, Texas, 1-4 October.

Wen, X. H., Durlofsky, L. J., and Edwards, M. G. (2003a). Upscaling of channel systems in two dimensions using flow-based grids. *Transport in Porous Media*, **51**(3), pp 343–366.

Wen, X. H., Durlofsky, L. J., and Edwards, M. G. (2003b). Use of border regions for improved permeability upscaling. *Mathematical Geology*, **35**, pp 521–547.

White, C. D. and Horne, R. N. (1987). Computing absolute transmissibility in the presence of fine scale heterogeneity, paper SPE 16011 presented at the SPE Symposium on Reservoir Simulation, San Antonio, Texas, 1-4 February.

Zijl, W. and Trykozko, A. (2001). Numerical homogenization of the absolute permeability using the conformal-nodal and the mixed-hybrid element method. *Tranport in Porous Media*, **44**(1), pp 33–62.

Appendix A

The StanLab plugin

The simulation, upscaling and grid generation techniques described in this work have been implemented into a software development platform called the "Gocad developer kit". This platform is used for the development and maintenance of Gocad itself but also for various functionality extensions called "plugins". This appendix describes the Gocad programming environment, the structure that guides the implementation of a plugin, and the specific functionalities that were implemented into the StanLab plugin.

A.1 Gocad programming environment

Gocad is a geomodeler with applications in several areas of earth sciences. It allows (among other tasks) one to perform structural modeling of reservoirs, seismic interpretation, integration of well-logs, geostatistical simulation and reservoir simulation grid generation. Although it is commercially distributed, Gocad has been designed not only to be extendable in a proprietary manner, but also to be open to other extensions performed by Gocad users, or by a third party. We now describe the Gocad software from the perspective

of a research application, client of a robust scientific library and visualization kernel.

A.1.1 Gocad products licensing

The Gocad program was written in the mid 1990s at the LIAD. It was written in C++ and supports several computer architectures. Gocad is now sold and maintained by Earth Decision Sciences (formerly known as TSurf). Gocad plugins may be of two types: commercial or research. Commercial plugins are sold through a Gocad licensing agreement while research plugins are usually developed at the LIAD[1] and made available to the consortium affiliate members. In the latter case, a Gocad license is still required to use the plugin, as Gocad acts as a kernel for the plugin application. Also note that research plugins may or may not come in a readable source-code format. In that case however, their functionalities are still accessible by concurrent plugins.

A.1.2 Gocad world, external developer perspective

The Gocad window application includes:

- a 3D view of the Gocad world that contains rendered views of Gocad objects,
- various menu bars and icons,
- a tree of Gocad objects that can be manipulated and passed as parameters of various functions.

[1] The StanLab plugin developed in this work is an exception

During the implementation of a Gocad plugin, two principal developments can be considered: the addition of new visualization capabilities, and/or the addition of new functionalities that use existing Gocad objects[2].

Several object types (*e.g.*, wells, surfaces, geometrical constraints) are already implemented in Gocad. However, new objects may be created for which "views" or aspect attributes may be specified. These objects enrich the set of Gocad object types and can be displayed and edited. This can be seen for instance in the TopoLab plugin where polyhedral cells (not present in the Gocad kernel) and associated properties can be visualized and edited. This important aspect of the Gocad programming environment is used in this work via the absorption of the TopoLab functionalities in the StanLab plugin.

The addition of new functionalities to the Gocad interface can be performed in a rather efficient way if basic guidelines are followed. New functionalities are made available through a sequence of programming steps that involve the creation of different layers of interface. As described further in this appendix, existing standalone programs (such as a flow simulator) can be called from the Gocad window using an appropriate data format conversion (filter). Also, new functionalities (such as streamline simulation on structured grids and upscaling techniques) can be implemented using existing Gocad objects. The principal benefit is that simulation or upscaling results can be displayed on Gocad grids, saving the usual postprocessing of simulation results files.

A.1.3 Programming structure

Developing applications based on the Gocad environment has three principal advantages: (a) it provides a large number of routines and data structures conveniently grouped into

[2]By Gocad object, we mean objects that the user manipulates through the 3D view, not internal programming object types

libraries, (b) it guides the application programming using different levels of interfacing that allow one to create graphical input of parameters as well as display of results, and (c) it allows for cooperative developments via standardized communication between plugins.

Programming layers

Gocad uses several layers of programming interface which allow for different levels of interaction with the elements of the program. For instance, a routine accessible from the graphical user interface offers a different number of parameters than the programmer may have originally provided. Between the user and the programmer levels, there is therefore a "stack" of interfaces that simplifies and partially specifies the parameterization for a particular functionality. This not only appears as a filter that hides some of the complexity of the routines to the user, but also expands the possible applications of a functionality as there are a number of points of entry to the functionality. To each level of entry may correspond a different "client".

Interface	Client
Graphical interface	User
Application programming interface (API)	Customization of the plugin
Library interface	Plugin developer

The following subsections describe the layers of programming interfaces from the highest to the lowest level.

Command line interpreter

The command line interpreter (CLI) and graphical command line interpreter (XCLI) form the highest level of interaction with the Gocad kernel and plugins. The CLI consists of

a string parser that transforms a user message into a parameterized Gocad action. This is useful for keeping a log of the commands during a Gocad session where each of the user inputs can be summarized as a series of character strings that can be fed back to Gocad during a later session, or to automate a series of actions that need to be repeated in sequence. This aspect allows the user the highest level of Gocad programming and is used in particular to generate "workflows". The XCLI is the graphical counterpart of the CLI. It is comprised of dialogue boxes, pull-down menus and contextual help. It can be viewed as a convenient proxy for generating strings interpretable by the CLI. Because the write-up of the CLI and XCLI routines is rather tedious and repetitive, Gocad provides its own language for generating the CLI and XCLI source code. This language, called gencli, allows one to generate CLI, XCLI codes and the appropriate "wiring" in an expedient and automated fashion.

Gocad **application programming interface**

The Gocad application programming interface (gapi) is a set of functions that can be called from the client interface. The structure of this interface is very simple. Functions that belong to the same category are grouped within the same subsystem or "namespace"[3]. Functions of the gapi may create, destroy or edit Gocad objects, thus updating views in the principal application window. Within the layered structure of the programming interfaces, the function calls tend to go "downward", *i.e.*, functions tend to use other functions from the same level or lower. Plugin routines (non-gapi) may call functions from the gapi, though this tends to be a poor programming design. For instance, the streamline simulator may create (stream) lines to be drawn on the screen using the gapi function `iline::create_iline_from_points(...)`.

[3]This is achieved in C++ by the implementation of "static" members of a common class

Gocad **library programming**

This is where most of the actual plugin programming tends to go. At this level, no reference is made to the Gocad session or objects. Many of the data types that are defined at this level can be used at all locations in the plugins. For instance, a list or an event-notification mechanism can be used in graphical or reservoir simulation routines.

A.2 StanLab Plugin functionalities

This section describes the functionalities available in the StanLab plugin via the graphical user interface. These functionalities fall into three main categories: structured grids, upscaling, and unstructured grids. Because these functionalities are directed toward the creation of an upscaled unstructured reservoir simulation model, the simulation-model data-structure is also discussed.

A.2.1 Structured grids

Structured grids are the starting point of the unstructured grid generation methodology proposed in this work. For that purpose, the following operations need to be performed on any given structured grid:

- reference single phase and tracer flow simulations,
- generation of flow information maps,
- single phase flow calculations for upscaling permeabilities.

Structured streamline flow simulation

Tracer flow simulation may be performed on a Gocad structured grid (called a Voxet) for the purpose of (a) calculating a water cut that will serve as a reference solution to be compared with the solutions obtained on unstructured grids, and (b) generating information maps such as flow-rate and/or total breakthrough-time maps. Also, the display of some of the streamlines used in the calculation may be beneficial to understanding the overall flow pattern. The function "Perform Tracer Flow Simulation", accessible through the Voxet Property menu, allows one to perform such a simulation for simple types of boundary conditions and creates the desired results as new properties attached to the Gocad Voxet model. The dialogue box gives the following options:

- select the Voxet,
- select the type of boundary conditions (left-to-right, front-to-back or top-to-bottom) and the value of the pressure drop across the model,
- select the Voxet property fields to be created (flow-rates, time of flight, total time of flight, and the fraction of the streamlines traced to be output).

Note that in the current version of the plugin, the resulting fractional flow curve is not displayed as a Gocad window but is rather output as a text file in the application current directory. The file contains three columns for each of the streamlines traced sorted by increasing breakthrough time: the breakthrough time (in PVI), the associated flow rate, the cumulative rate fraction (between 0 and 1).

Furthermore the option of performing a single phase flow simulation on the grid is offered. This calculation is faster than the tracer simulation and also includes the additional option of using periodic boundary conditions (with fixed pressure drop along one direction).

Flow information maps

After a streamline simulation, the Voxet property fields generated can be used to create flow information maps.

For a flow-rate map, one possibility is to average flow rates obtained for different boundary conditions. The resulting map is then obtained via the Gocad property menu:

1. create the flow rate map using "Create New Property"
2. average the flow rates using the "Apply Script" function and

$$\bar{v} = \alpha|\mathbf{v}^x| + \beta|\mathbf{v}^y| + \gamma|\mathbf{v}^z| \tag{A.1}$$

   where α, β and γ are scaling coefficients and $\mathbf{v}^u$ is the velocity vector obtained by applying a pressure drop in the $u \in \{x, y, z\}$ direction.

For the total breakthrough time, a complication may arise due to the fact that not all simulation blocks are crossed by a streamline. A possibly large fraction of the cells therefore does not have a value. The solution we suggest is to:

- perform the streamline simulation in the $u+$ direction,
- perform the streamline simulation in the $u-$ direction,
- merge the two property fields,
- interpolate the values at unsampled cells using the Gocad kriging capabilities.

Map calibration

Statistics on the information map may be obtained within Gocad using the "Spatial Data Analysis" wizard. This tool provides histograms, percentiles and directional variograms.

Modifications of the map may again be obtained using Gocad property scripts, but one may also use the end-point and shape-factor parameterization of the resolution constraints available from TGridLab, as discussed in Section A.3.

A.2.2 Upscaling

Upscaling capabilities were implemented in the context of unstructured polyhedral grids. As polyhedral grids are not accounted for in the current version of Gocad (2.0.7), the upscaling functionality requires the use of the TopoLab plugin. In our context, polyhedral grids would typically be obtained as the dual of a tetrahedral mesh, although this is not necessarily the case. Indeed, Cartesian or corner-point grids can be converted to the TopoLab polyhedral data structure (using existing menus) and our upscaling routines can then be applied to all types of grids.

Polyhedral grids are called “PSolid” in TopoLab. The upscaling routines were added (in StanLab) under the PSolid menu. Two types of upscaling are considered: transmissibility upscaling, which calculates a two-point transmissibility coefficient associated with any polygonal face of the PSolid, and permeability upscaling, which allows for the calculation of a full tensor for any polyhedral cell. For a given cell, the procedure fits an ellipsoid through points of the surface and extracts a 3D subset of the underlying Cartesian model (extended region). The extended region is used to perform the single phase flow calculations, while the ellipsoid is used to approximate the average pressure gradients and flow rates on the polyhedral cell. The flexibility of the PSolid grid property management allows values to be associated with any dimension of the grid: points, edges, faces or volumes.

A.2.3 Unstructured grid

Two types of unstructured grids can be handled in StanLab: (a) simplicial unstructured grids comprised of tetrahedra and for which the TGridLab plugin offers advanced grid generation capabilities, and (b) polyhedral grids formed in the TopoLab plugin that can be converted from other grid types (such as structured or tetrahedral grids) or generated as the dual of another grid. We now discuss the grid generation using TGridLab, the unstructured grid flow simulation using the FlowLab library, and the creation of simulation runs from the StanLab interface.

The TGridLab plugin

The TGridLab plugin generates simplicial grids constrained to user defined data. There are two types of constraints: topological and grid resolution constraints.

Topological constraints such as internal faults, horizons or reservoir boundaries are provided by a constrained structural model. The constraints of the model represent the type of contacts between intersecting surfaces. For instance in the case of a “Y” fault, the branches of the fault rarely exactly intersect due to geometrical approximations. It is critical not to insert any artificial gap between faults, as the gridding algorithm will honor the fictitious gap geometry, introducing many unnecessary grid nodes. Whenever surfaces intersect or are in contact, explicit constraints must therefore be set in the model. In the case of the “Y” fault, a “Border on Surface” constraint “attaches” the antithetic (small) fault to the synthetic (large) one. Also, to ensure that the contact between the faults covers the entire width of the fault, an additional constraint must be set on the fault contact line. This constraint is of type “Border Extremity on Border”. The construction of the surface constraints may be automated to a certain extent when using a Gocad “Reservoir Model”

object. Indeed, the creation of an intermediate reservoir model[4] from intersecting surfaces automatically introduces constraints based on distance criteria. The updated surfaces can then be used by the TGridLab plugin, as we now explain.

To construct a simplicial grid using TGridLab, a "Wireframe Model" must first be constructed. An instance of a wireframe model is a TGridLab object that synthesizes the grid information for objects of different dimensions (volumes, surfaces and lines). It also embeds the topological constraints to ensure that gridded objects are consistent, *i.e.*, that grids conform to their constraints. For instance, two intersecting surfaces may be independently triangulated but their intersection must have a unique 1D discretization. The wireframe model ensures this consistency. Once the wireframe model is constructed, an initial constrained tetrahedral grid can be constructed. This is done by first choosing a 1D discretization for lines, then a target triangle size for each surface, and finally a target tetrahedron size for volumes. Finally, resolution constraints may be specified and the grid may be adapted. The resolution constraint is provided by a property server that may be that of a surface (TSurf), a tetrahedral grid (TSolid) or more generally of a structured grid (Voxet).

Resolution constraints may be used to control the node density of either the surfaces that were imbedded in the wireframe model or of the 3D grid itself. This is done via the Wireframe Model menu "Add Resolution Constraints". We can then adapt the surfaces using the 2D constraints (this automatically updates the tetrahedrization). Finally the 3D grid is adapted using the 3D constraints, and the last operation exports the tetrahedrization to the Gocad window, converting it from the wireframe to a standard Gocad TSolid format.

Note that the TGridLab plugin contains many more functionalities than those discussed in this appendix.

[4]The Gocad "Reservoir Model" should not be confused with the StanLab "Reservoir Simulation Model" described later in this appendix

From Flex **to** FlowLab

StanLab allows us to perform basic flow calculations on unstructured grids. The functionalities of Flex were carried over to FlowLab, a stand alone library package that can be linked to the StanLab plugin. The new libraries introduce several enhancements. The use of the Standard Template Library data types and algorithms was generalized. New boundary conditions were added such as fixed BHP/block pressure for the well controls or a numerical aquifer. Also, FlowLab is able to perform 3D streamline simulations. In summary, StanLab allows one to run unstructured grid simulations either as a stand alone program that reads a text-based input, or directly from StanLab where data for the grid, permeability field, transmissibility coefficients and wells are converted from the Gocad to the FlowLab data type. The data structure that allows us to create a flow simulation model is now explained.

A.2.4 Simulation data structure

In order to integrate the assessment of the unstructured grid quality to the adapted grid generation workflow, we implemented the capability of running a tracer flow simulation directly from the StanLab window. This option calls routines from the FlowLab library. Hence, all data needed for the simulation must be acquired or generated by StanLab and transferred to FlowLab. The data we need to transfer are:

- the primal grid, comprised of
  - the coordinates of vertices of the grid
  - the topology (connections between the nodes)
- a distribution of rock properties given as a list of:

  - 3D points
  - associated permeability
  - porosity

- the upscaled transmissibilities which, in the case of the two-point flux approximation, are given as a list of triplets that contains:
  - the index of the two connected nodes i, j
  - the associated transmissibility coefficient $T_{i,j}$
- the fluid PVT properties
- the boundary condition definitions (wells and aquifers)
- various controls such as time stepping, well conditions, *etc*.
- some tuning parameters

Some of these data can be automatically generated by StanLab such as the PVT data for a tracer flow; others are entered by the user (*e.g.*, wells) or are stored as a result of an internal calculation (transmissibility upscaling for example). The reservoir simulation data are stored in the following three macro structures:

- 'Simulation Grids'
- 'Boundary Conditions'
- 'Rock Model'

We now discuss each of these structures in more detail.

Simulation Grids

This structure handles the primal and dual grids and their correspondence. This complexity is motivated by the fact that we want to be able to pass connectivity lists to the simulator and because we need to handle two different grid-types. While the upscaled transmissibility coefficients are calculated on the faces of the dual grid, the simulator needs to know between which of the primal nodes i, j the flow connection is to be created. This structure therefore keeps track of the correspondence between a polyhedral cell and the dual vertex it originated from. Note that it is possible to create an upscaled permeability tensor field on any polyhedral grid, but that transmissibility upscaling requires that both the primal and dual grids be stored in the Simulation Grid structure.

Boundary Conditions

The "Boundary Condition" structure stores the list of the reservoir features that interact with the reservoir model. Each feature (well or aquifer) has three different aspects (or views): a geometry, a control logic, and a numerical model.

The geometry of a boundary condition can have different specifications. Specifically, it can be a 2D or 1D object and can be represented by a discretized object or an analytical expression. Below is a table that summarizes these cases:

		Dimension	
		1D	2D
Parameterization	Discrete	a well	a 3D aquifer
	Analytical	2D aquifer	(not used)

Note that the geometry of the boundary condition is independent of the grid. The same

structure is therefore used for storing the simulation data of both structured and unstructured runs. The simulator introduces the connection between the general geometry of the boundary condition and the grid provided. The control of the boundary condition allows relatively little flexibility: fixed BHP or total flow rate for wells, and fixed pressure for the aquifers. The fixed pressure aquifer was used to simulate face-to-face flow in the results presented in this thesis.

The numerical model view specifies the way the boundary condition interacts with the reservoir numerical model. In the case of a well, it is through a well index, in the case of a fixed pressure aquifer, the block pressure unknowns are "linked" to a fixed pressure value. When multiphase flow is considered, it is this part of the model that decides how the mobility of the fluid should be calculated (in the case of an injecting well for instance), or what fluid should be injected into the reservoir (in the case of a fixed pressure aquifer).

Rock Model

The Rock Model stores all the results from the upscaling procedures. This includes the porosity values, permeability tensors, and transmissibility coefficients. These fields are stored in the embedding of the PSolid (either on the 3D or 2D cells).

A.3 Grid generation workflow

This section describes a step by step procedure to generate an adapted unstructured simulation run using StanLab. The three principal steps are the data preparation, adapted gridding, and reservoir simulation modeling.

A.3.1 Data preparation

We now describe briefly how to construct the fine model property distribution and structural information.

Voxet

The Gocad structured grid (Voxet) is created using "Voxet→ New→ From Step Vectors". Properties are assigned using Gocad geostatistical capabilities. First, a variogram is constructed using the "Application→ Spatial Data Analysis", then we perform a kriging using "Geostatistics→ Unconditional SGS". Finally, we create the basic structural information using the following steps:

1. "Surface→ Create From Object Box" (use the Voxet) creates one surface with 6 parts
2. "Surface→ Create From Surface Part" creates 6 surfaces. At this stage, no constraints exist between them
3. "Surface→ Model3D→ New From Surfaces" creates copies of the surfaces on which the appropriate constraints are introduced

Flow information map

We then perform tracer flow simulations through StanLab menus: "Voxet→ Property→ Flow→ Tracer Simulation" using the permeability/porosity model created earlier. The resulting flow property fields are attached to the Voxet, while the fractional flow curve is saved in an ASCII file. Running similar single phase flow problems using different boundary conditions, we generate an average flow-rate map using Equation A.1 in the menu "Voxet→ Compute→ Apply Script on Object".

A.3.2 Adapted Gridding

We now use the TGridLab functionalities to generate an unstructured grid adapted to the flow rate information. We follow the steps below:

1. create the frame model via “Framemodel→ Create from Surfaces” (or “Model3D”)

2. initialize the 1D, then 2D, then 3D tesselations using “Framemodel→ Create Tesselation. . . ”

3. install the grid resolution constraint via “Framemodel→ Constraints→ Add 3D Constraint”. The window application allows one to choose the conversion between the flow-rate values and the target grid resolution

4. update the tesselation using the resolution constraint “Framemodel→ Refine Grid→ With Constraints”

5. export the grid under a format that Gocad can handle (TSolid): “Framemodel → Export→ TSolid from Tesselation”

Statistics on the number of tetrahedra and vertices can be obtained in the “Info” property of the TSolid created.

A.3.3 Reservoir simulation model

Instead of creating the dual grid directly from the “Psolid→ Create from Dual” menu of StanLab, we create a Reservoir Simulation Model. The steps are as follows:

1. create a TFlowSimulation from “Reservoir Simulation→ Create from Primal Grid”

2. compute the dual grid using “Reservoir Simulation→ Update Dual Grid” (the corresponding PSolid dual grid is created in the Gocad world)

3. perfom the desired upscaling (permeability and optionally transmissibility; permeability may still be required for the well index calculations)

4. add boundary conditions

5. create a tracer simulation run “Reservoir Simulation→ Export→ Create a Simulation Deck”

The user is then prompted for a directory location to which all the data are transferred to execute the simulation. A default fluid model file is also generated for the tracer flow case. An alternative to the last step is to directly execute the simulation run from the StanLab window. However, this option is limited in the current version of the plugin since the results are not transferred back to the dual grid. This extension however can be achieved easily.