Streamline methods with application to flow and transport in fractured media

Doctor Scientiarum Thesis

Håkon Hægland

University of Bergen

2009

Håkon Hægland
Department of Mathematics
University of Bergen
Joh. Brunsgt. 12
NO-5008 Bergen
Norway
hakonh@math.uib.no

ISBN 82-308-0034-0
University of Bergen, Norway
Printed by Allkopi, Bergen

Preface

The work in this thesis has been done as part of a PhD program at the Department of Mathematics, University of Bergen, Norway.

The thesis has been organized in three parts: Part I discusses background material and a general framework that explains the connection between the included papers, Part II includes submitted and published research papers, and Part III includes two additional supporting co-authored research papers.

All the papers are related to the field of streamline simulation within reservoir simulation. Papers A-E are concerned with streamline tracing, whereas Paper F considers the porous media single phase flow equation in a discretely fractured media.

The included papers in Part II are summarized as follows:

Paper A: Hægland, H., Dahle, H.K., Eigestad, G.T., Lie, K.-A., Aavatsmark, I. Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Advances in Water Resources*, 30(4), 1027-1045, 2007.

We develop a new interpolation scheme on general hexahedral grids. The main advantage of this method is that it can reproduce a uniform flow field, and allows accurate streamlines to be traced. In fact, as far as we know, this method is the method with the lowest degrees of freedom with this property. A disadvantage is that the method not generally produce divergence free velocity fields, see Paper B.

Paper B: Nordbotten, J.M., Hægland, H. On reproducing uniform flow exactly on general hexahedral cells using one degree of freedom per surface. *Advances in Water Resources*, doi:10.1016/j.advwatres.2008.11.005, 2008.

We recognize that an interpolation method that both reproduces uniform flow and is divergence free and only uses one degree of freedom per face does not exist for a general hexahedron. This means that the method developed in Paper A is not divergence free. A fix is suggested in Chapter 5.1.

Paper C: Hægland, H., Dahle, H.K., Lie, K.-A., Eigestad, G.T. Adaptive stream-

line tracing for streamline simulation on irregular grids. *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, 2006.

A fast method is proposed for tracing streamlines using the velocity field derived in Paper A.

Paper D: Hægland, H., Dahle, H.K., Eigestad, G.T., Nordbotten, J.M., Celia, M.A., Assteerawatt, A. Streamline methods on fault adapted grids for risk assessment of storage of CO_2 in geological formations. *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, 2006.

In this paper, methods for streamline tracing on faulted and fractured grids are proposed.

Paper E: Hægland, H., Assteerawatt, A., Helmig, R., Dahle, H.K. Simulation of flow and transport in discrete fracture-matrix system II. Efficient and accurate streamline approach. Submitted to *Water Resources Research*, 2008.

A flux post-processing technique is developed that allows streamlines to be traced when a lower-dimensional method is used to solve the flow equation.

Paper F: Hægland, H., Assteerawatt, A., Dahle, H.K., Eigestad, G.T., Helmig, R. Comparison of discretization methods for flow in a fracture-matrix system. Submitted to *Advances in Water Resources*, 2009.

A comparison of different discretization methods for flow is done in the context of a 2D discrete fracture model.

As all the papers are collaborative work, some remarks about my contribution are necessary. For Papers A, C, D, and F, I was the first author and my contributions were correspondingly. In particular, in Paper A, I came up with the CVI interpolation scheme. In general, my contribution in the previous mentioned papers were development, implementation and numerical testing of the methods. In Paper B, we prove a theorem that is relevant for both streamline tracing and mixed finite-element methods. In this paper, I was responsible for placing the result in a general setting including the literature review. In Paper E, I was involved in the development, implementation and numerical testing of the methods.

Further, this thesis contain the following coauthored supporting papers, (not considered as the main part of the thesis):

Paper G: Kippe, V., Hægland, H., Lie, K.-A. A method to improve the mass balance in streamline simulation, SPE 106250. Presented at the *SPE Reservoir Simulation Symposium*, 2007

Paper H: Assteerawatt, A., Helmig, R., Hægland, H., Bárdossy, A. Dahle, H.K. Simulation of flow and transport processes in a discrete fracture-matrix system I. geostatistical generation of fractures on an aquifer analogue scale. Submitted to *Water Resources Research*, 2008.

The relation of these papers to the main part of the thesis is further discussed in Chapter 6.

Acknowledgments

First of all, I want to thank my supervisors Helge Dahle and Knut-Andreas Lie for their patience, help, and extended critiques of my work during my PhD project. A special thanks to my main supervisor Professor Helge Dahle for constantly pushing me forward and arranging several research stays abroad and conference participations for me. I am also very grateful to Geir Terje Eigestad and Ivar Aavatsmark for help and encouragements during this work. My collaboration with Geir Terje Eigestad has been of particular value in finalizing parts of this thesis.

I consider the periods abroad as very important for my PhD work. These stays have been helpful and instructive in setting my research into perspective, learning to communicate with other researchers, and giving me new insights and establishing contacts. First, I appreciated an extended visit of six months at Princeton University; I want to thank Professor Michael A. Celia for offering me this great opportunity to work in an excellent and stimulating research environment. Then, I am deeply grateful to Professor Rainer Helmig at the University of Stuttgart, for the four months I spent there in a sequence of five separate short stays. I want to thank him specially for his great hospitality, for his positive, encouraging attitude, and pure interest in my work, which stimulated me to further accomplishments. During my stay at Princeton, I also appreciated a one week memorable stay at Stanford University, and I want to thank Margot Gerritsen for giving me this opportunity and for being an excellent host.

In addition, I had several short stays at SINTEF ICT hosted by Professor Knut-Andreas Lie. I want to thank him for those periods. I also want to thank him for the excellent winter schools at Geilo, and the nice discussions we had there. Then, I would like to thank Jan Nordbotten for all comments, advices, discussions, and for being a great example for research in general during my PhD-period both in Princeton and in Bergen. At last, thanks to my fellow students Anongnart

Assteerawatt, Vegard Kippe, and Birgitte Eikemo. I have benefited greatly from collaborating with you!

Håkon Hægland
Bergen, February 2009

Contents

I Background Material **1**

1 Introduction **3**
1.1 Structure of the thesis 6

2 Conceptual model **7**
2.1 Models for a porous medium 7
2.2 Models for fractured reservoirs 9
2.3 Models for a fracture 13

3 Mathematical model **17**
3.1 Single phase flow 18
3.2 Single phase, tracer transport 18
3.3 Two phase flow 19
3.3.1 Streamline formulation 21

4 Numerical model **23**
4.1 Finite volume methods for elliptic equations 25
4.1.1 Vertex-centered methods 26
4.1.2 Cell-centered methods 28
4.1.3 The box method 30
4.1.4 The MPFA O-method 33
4.2 Flux recovery 38
4.2.1 Control volumes with single fractures 39
4.3 Streamline tracing 44
4.3.1 Short literature review 47
4.3.2 Introduction: Tracing on Cartesian Grids 48
4.3.3 Streamline and Velocity in Curvilinear Coordinates 51
4.3.4 Extending Pollock's Method to Irregular Grids 52
4.3.5 Extending the SFM and EFM Methods to 3D 54
4.3.6 Reproduction of Uniform Flow 55
4.4 Corner Velocity Interpolation 56

4.4.1 Interpolation in 2D 56
4.4.2 Extension to 3D 57

5 Ongoing and further work **61**
5.1 Improved CVI method 61
5.2 Vertical averaging and normal lines 63

6 Supporting material **67**

Bibliography **69**

II Included Papers 83

A Improved streamlines and time-of-flight for streamline simulation on irregular grids

B On reproducing uniform flow exactly on general hexahedral cells using one degree of freedom per surface

C Adaptive streamline tracing for streamline simulation on irregular grids

D Streamline methods on fault adapted grids for risk assessment of storage of CO_2 in geological formations

E Comparison of cell- and vertex centered discretization methods for flow in a two-dimensional discrete fracture-matrix system

F Simulation of flow and transport in discrete fracture-matrix system: II. Efficient and accurate streamline approach

III Supporting material

G Simulation of flow and transport in a discrete fracture-matrix system I. geostatistical generation of fractures on an aquifer analogue scale

H A method to improve the mass balance in streamline methods

Part I

Background Material

Chapter 1

Introduction

The main theme of this thesis is fast and accurate simulation of fluid flow and transport in porous media. In particular, we focus on problems leading to irregular grids including fractures. Flow simulations are carried out by a finite volume method that ensures proper treatment of heterogeneities. The velocity recovered from the flow simulations are then used to trace streamlines that are used to obtain fast and accurate transport simulations. Transport simulations are used for understanding and predicting the geological system under consideration, using tools like break-through curves, arrival time and storage behavior. The outcome of transport simulations strongly depends on the distribution of velocities in discontinuous features like fractures [51]. Hence, the success of a transport simulation is closely related and linked to the efficiency and accuracy of a) the numerical scheme for the flow equation, and b) the method used to trace streamlines. In particular, we consider streamline tracing for irregular hexahedral grid cells in 3D and for discrete fractures in 2D.

Important application areas are groundwater pollution, improved oil recovery, and geological sequestration of CO_2. Groundwater is used as drinking water, and it is used in agriculture and industry. The supply of groundwater is being limited by an increasing human population, by pollution, and by climate change. Contamination may arise from, e.g., radioactive waste in repositories in deep geological formations, or from leakage from landfills. The understanding of the flow of water in the subsurface is important for decision making and risk analysis.

An ever expanding supply of oil and natural gas has been vital to continued economic growth. Enhanced recovery techniques are used to increase the amount of oil that can be extracted from a reservoir, and are becoming increasingly important as the world's oil reserves are diminishing. It is important to understand the fluid flow processes in petroleum reservoirs sufficiently well to be able to optimize the recovery of hydrocarbon.

Environmental concerns is also becoming more important in the oil industry.

Over the past century, human activities such as the burning of fossil fuels, have contributed to a large increase in the amounts of carbon dioxide (CO_2) in the atmosphere. It is now commonly accepted that this rapid increase in atmospheric CO_2 has resulted in a global warming and a following climate change. A variety of strategies have been proposed to reduce CO_2 emissions and remove carbon from the atmosphere in order to mitigate the potential effects of climate change. One option is CO_2 sequestration, where the CO_2 produced from a power plant is separated, captured and then stored in a deep underground geologic formation in such a way that the CO_2 will remain permanently deposited. The potential environmental consequences and risks to public safety with such a procedure are currently insufficiently studied through systematic research. Leakage of CO_2 from underground fractures or abandoned wells can be a risk to human health, and make the CO_2 sequestration inefficient.

The aim of this thesis is to help to better understand these problems, by building a numerical model for fluid flow in porous medium that can be solved on a computer, and subsequently be used to predict, e.g., the performance of an oil reservoir, the risk associated with a CO_2 injection, or the spreading of pollution in an aquifer, given certain initial and boundary conditions. A common feature of these application areas is that they share (more or less) the same physics, and can be formulated mathematically using the same balance laws.

Geological heterogeneity that controls flow is present in all rocks [59]. The depositional characteristics that give rise to permeable formations, and the complex diagenetic processes taking place afterwards, create important heterogeneous features such as bedding planes, fractures, and faults [118].

Fractures are mechanical breaks in rocks involving discontinuities in displacement across surfaces. Fractures occur on many scales, with different aperture and permeability. They may be open channels in the rock, or they may be filled with clay or debris. Open fractures are often preferential flow paths [88].

Rock heterogeneities have strong impact on fluid displacement patterns because they define preferential flow paths in underground permeable formations. The efficiency and accuracy of a numerical simulation model is greatly controlled by the ability to understand and represent this heterogeneity [51].

Since the pore geometry of a reservoir is commonly inaccessible to direct investigation, a continuum approach is used where rock and flow properties are transformed from a microscopic level to a macroscopic one [23]. The transformation involves an average over a representative elementary volume (REV) [23, 26, 27]. With this description, the pore structure is replaced by continuous properties like porosity and permeability. The permeability tensor is function of the position in the medium, and quantifies the ability of the rock to transmit fluid.

To deal with different fracture distributions, matrix properties and limited computing resources, a number of modeling approaches have been developed for

fracture-matrix systems. These modeling approaches can be formulated in deterministic and stochastic frameworks, and are traditionally divided into two rough classes: discrete fracture models and continuum models [29].

Continuum models are based on the spatial averaging approach [25] assuming the REV to cover both fractures and matrix. Hence, individual fractures are not explicitly treated, but expressed through volume averaged behavior of many fractures. The REV concept can be used if a formation contains a dense network of highly interconnected fractures. If a REV can only be defined at a scale similar to the problem of interest, as is the case for poorly connected networks, or if a network consists of fractures with no characteristic size limit, then the REV approach is inappropriate [29].

Discrete fracture models allow quantification of many flow and transport phenomena that are not adequately captured by continuum models [57]. An advantage of the discrete fracture approach is that it can account explicitly for the effects of individual fractures on fluid flow and transport. Applications of the discrete fracture model to field scale problems have been limited due to insufficient access to accurate data and computational constraints due to the huge amount of data required to represent the discrete features. However, the discrete fracture model have gradually become more practical, as a consequence of increased computer power and improved simulation and characterization techniques.

On the basis of the conceptual model for the reservoir heterogeneity, a mathematical model for flow and transport is developed. The mathematical model of the physical behavior of the system describes how the fluid distributes itself in the porous medium given certain initial and boundary conditions [70]. The model is based on the continuum approach (REV). It consists of a set of partial differential equations derived from the principle of mass conservation in combination with the momentum equations, known as Darcy's laws [15].

When more than one fluid is present, e.g., oil and water, the different fluids are considered separate phases, each of which occupy a fraction of the REV. The fraction of a given phase at a given point is called the saturation of the phase. The coefficients of the flow equations depend on the porosity, permeability of the porous medium, and among others, the densities and viscosities of the fluids. The unknowns in the flow equations are the pressures and the saturations. When pressure is known, the fluid velocities can be calculated from Darcy's law.

With general initial and boundary conditions, the flow equations cannot be solved analytically. Instead, a discrete form (numerical model) of the equations is developed that can be solved on a computer. The flow equations may be formulated in different ways, giving rise to different numerical solution methods. For the flow of two immiscible phases, assuming incompressible fluids and rocks and neglecting capillary pressure, one such formulation is given by an elliptic equation for the pressure and a hyperbolic equation for the saturation. An efficient

numerical solution procedure for these equations is the implicit pressure explicit saturation (IMPES) method [128]. The IMPES method can be further speeded up by a coordinate transformation in terms of the time-of-flight coordinate along streamlines. In this way, the saturation equation is reduced to a set of 1D hyperbolic equations along streamlines [22].

The numerical model involves a discretization of the flow equations. The reservoir is subdivided into grid of grid cells corresponding to regions of constant material properties. For a rectangular domain, with heterogeneities aligned with the grid boundaries, the grid cells can be rectangular parallelepipeds. Such a grid is called a logical Cartesian grid. However, the usual case is that the domain boundary is irregular and internal layers or fractures are not aligned with each other or with the boundary. Irregular and unstructured grids may then be used to better fit the grids cells to the boundaries of the reservoir and to the internal media discontinuities.

1.1 Structure of the thesis

In Chapter 2, we describe the conceptual model for a porous medium, a fracture-matrix system, and a fracture. In Chapter 3, we develop the mathematical model for flow and transport in porous medium, that may or may not contain fractures. Then, Chapter 4 describes the numerical methods we have used to solve the mathematical models for certain initial and boundary conditions. Further, Chapter 5 outline some ongoing and further work, and Chapter 6 gives a brief introduction to the supporting material. Part II includes the research papers that constitutes the main part of this thesis, whereas Part III includes the supporting papers.

Chapter 2

Conceptual model

With a model we mean a simplified version of the real system that approximately simulates its behavior [26]. The first step in the modeling process is the construction of a conceptual model of the system and of its behavior. This step includes a verbal description of the system's composition, the physical phenomena that take place and the mechanisms that govern them, as well as the relevant properties of the medium in which they occur, all subject to the required output of the model.

The description takes the form of a set of assumptions, subjectively selected by the modeler, to express his understanding and approximation of the real system and processes taking place within it [26].

2.1 Models for a porous medium

Phenomena of transport in porous media are encountered in many engineering disciplines. Civil engineering deals, for example, with the flow of water in aquifers, the movement of moisture through and under engineering structures, and transport of pollutants in aquifers. Reservoir engineers deal with the flow of oil, water and gas in petroleum reservoirs.

A porous media is occupied by a persistent solid face, called matrix, and a void space occupied by a single fluid, or by a number of fluid phases, e.g., gas, water and oil. A phase is defined as a chemically homogeneous portion of a system. There can be only one gaseous phase in the system, as all gaseous phases are completely miscible. We may however have more than one liquid phase in a system. Such liquid phases are referred to as immiscible fluids. A component is part of a phase that is composed of an identifiable homogeneous chemical species [82].

The outcome of any given phenomenon in a porous system depends on several length scales over which the system may or may not be homogeneous. In general,

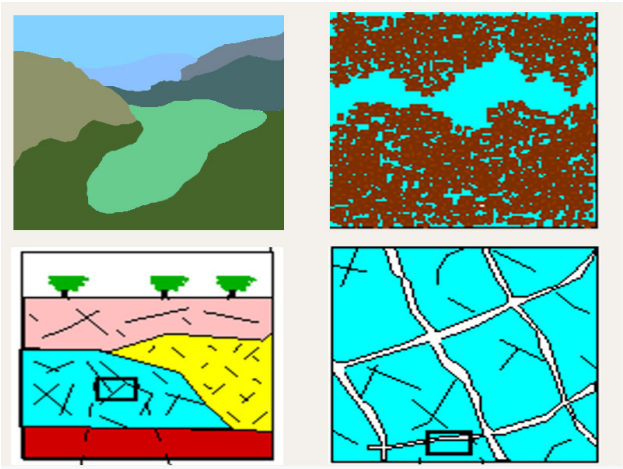


Figure 2.1: Different scales, modified after Assteerawatt [14]. Top left: the gigascopic scale, bottom left: the megascopic scale, bottom right: the macroscopic scale, top right: the microscopic scale.

the heterogeneities of a porous medium can be described at mainly four different length scales [79, 142] :

- The *microscopic* heterogeneities are at the level of the pores or the grains.
- The *macroscopic* heterogeneities are at the level of core plugs. Such heterogeneities are found in every well with property values varying widely from core to core.
- The *megascopic* heterogeneities are at the level of the entire reservoirs which may have large fractures and faults. They can be modeled as a collection of thousands, perhaps millions, of cores, oriented and organized in some fashion, each having a average effective property.
- The *gigascopic* heterogeneities are encountered in landscapes that may contain many megascopic reservoirs, along with mountains, rivers, etc.

Note that the scale below the microscopic one is commonly called molecular scale, but it will not be important for this work since we will consider a continuum approach, see the next section.

Any realistic modeling of flow and transport phenomena in a porous medium has to include a realistic model of the medium itself. The model should depend on the type of porous media we deal with, and the given computational limitations. We must use models that are simple enough to allow for simulation of various flow and transport phenomena with reasonable computation time, while they also

contain the essential features of the porous medium of interest. We refer to [142] for a review of different models for porous media.

In principle, the mathematical model that describes the transport of an extensive quantity of a fluid phase through a porous medium domain, can be stated for every point within the considered phase. This description is said to be at the microscopic level, as the focus is on what happens at a mathematical point within a phase in the domain [25]. Although the transport problem can be stated as a well posed model, it cannot be solved at this level, since the detailed geometry of the surface that bounds the phase is not known and is too complex to be described. As a consequence, the description and solution of a flow and transport problem at the microscopic level is impossible.

To circumvent these difficulties, the transport problem is transformed from the microscopic level to a macroscopic one, at which the transport problem is reformulated in terms of averages of the microscopic values [23]. The average values are measurable quantities. We refer to this approach as the continuum approach.

In the continuum approach, the real porous medium domain, consisting of two or more phases, is replaced by a model in which each phase is assumed to be present at every point within the entire domain. Each phase, thus, behaves as a continuum that fills up the entire domain. We speak of overlapping continua, each corresponding to one of the phases, or components [23].

For every point within each of these continua, average values of phase and component variables are taken over an elementary volume centered at the point. The average values are referred to as macroscopic values of the considered variables.

In order to avoid that the averaged values depends on the averaging volume, we need a criterion that determines a range of volumes where the averaged values remain, more or less, constant. An averaging volume which belongs to that range is considered a representative elementary volume (REV).

If such a REV cannot be found for a given domain it cannot qualify as porous medium domain. A conceptual determination of the size of an REV for a domain is shown in Fig 2.2. The determination is based on porosity as a geometrical property of the porous medium.

2.2 Models for fractured reservoirs

Rocks in the earth's crust are in general fractured, i.e., broken up by joints and faults. Pressurized fluid is frequently present both in open fractures and in the pores of the rock itself [88]. Fractures may form in response to stress due to, e.g., the weight of the earth's crust, high fluid pressure, tectonic forces, or thermal

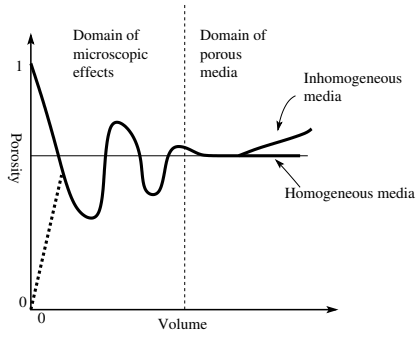


Figure 2.2: Definition of porosity and representative elementary volume. After Bear [23].

loading. Fractures also occur at many scales, from microscopic to continental. They may be "filled" with various minerals, such as calcite, dolomite, quartz of clay minerals, or they may be "open", in which case they may be filled with fluids under pressure.

Fractures are important in engineering, geotechnical, and hydrogeological practice. They can act as hydraulic conductors, providing pathways for fluid flow or barriers that prevent flow across them. In many rock masses, field-scale fluid flow takes place predominantly through fractures, rather than through the matrix rock itself. In some cases most of the flow may take place through a single fracture, whereas in other cases the flow occurs through an interconnected network of fractures. Many petroleum, gas, geothermal, and water supply reservoirs form in fractured rocks. Fractures can control the transport of chemical contaminants into and through the subsurface. Nearly half of all known hydrocarbon reserves are located in naturally fractured formations [117], as are most geothermal reservoirs [32]. Fracture flow is of importance in understanding and predicting the performance of underground radioactive waste repositories [158].

The choice of a model concept for the description of fractured media strongly depends on the scale of the problem, the geological characteristics of the area of investigation, and the purpose of the simulation. Bear [25] classifies various problems of flow and transport in fractured porous media as follows:

- **Zone 1: The very near field.** Interest is focused on flow and transport processes within a single, well-defined fracture, possibly with transport into the porous blocks that bound it.

- **Zone 2: The near field.** Flow and transport processes are considered in a relatively small domain, which contains a small number of well-defined fractures. The location and shape of the individual fractures are either deterministically defined or can be generated stochastically, based on statistical

information from the real system.

- **Zone 3: The far field.** On this scale, the flow and transport processes are regarded as taking place, simultaneously, in at least two continua. One continuum is composed by a network of large scale fractures and the other one by the porous rock. Mass of the fluid phase and its components may be exchanged between the two continua.

- **Zone 4: The very far field.** The entire fractured medium is considered as one single continuum, possibly heterogeneous and anisotropic in order to account for large scale geological layers and fault zones.

Hence, to set up models of systems with such varying characteristics, different model concepts are necessary. These concepts are also illustrated in Figure 2.3. The models can be classified into 1) continuum models, or 2) discrete fracture models, or 3) hybrid models, see e.g. [10, 20, 29, 51, 57, 93, 115, 119, 129, 141, 142]. The models differ in their representation of the heterogeneity of the fractured media, and whether they are formulated in a deterministic or a stochastic framework.

Continuum models [19, 25, 30, 45, 51, 93, 119, 141, 155] are based on the spatial averaging approach. The REV concept [23] is justified if a formation contains a dense network of highly interconnected fractures. If a REV can only be defined at a scale similar to the problem of interest, as is the case for poorly connected networks, or if a network consists of fractures with no characteristic size limit, then the REV approach is inappropriate [29]. Continuum models include the dual-porosity model (DPM) [19, 45], which has been widely used for large-scale simulations, especially in the oil industry. In this case the rock matrix form one continuum and the fracture network another, overlapping continuum [19, 155]. If one of these overlapping continua dominates all relevant aspects of flow and transport, the system acts as a single matrix- or fracture-dominated continuum. Otherwise, the model accounts for possible fluid and solute migration from one continuum to the other under a pressure and/or concentration difference between the two. If the fluid and mass transfer between the two is fast in comparison to flow and transport through the rock, one may consider the two continua to be at equilibrium and treat the rock as an equivalent effective (single) matrix-fracture continuum. In the absence of such equilibrium, it is common to adopt a dual porosity model in which the matrix acts as non-conducting storage reservoir; the fractures form a conducting medium with negligible storage capacity; and the transfer of fluids (or solutes) between these two overlapping continua is linearly proportional to the pressure (or concentration) differential between them at each point in space-time.

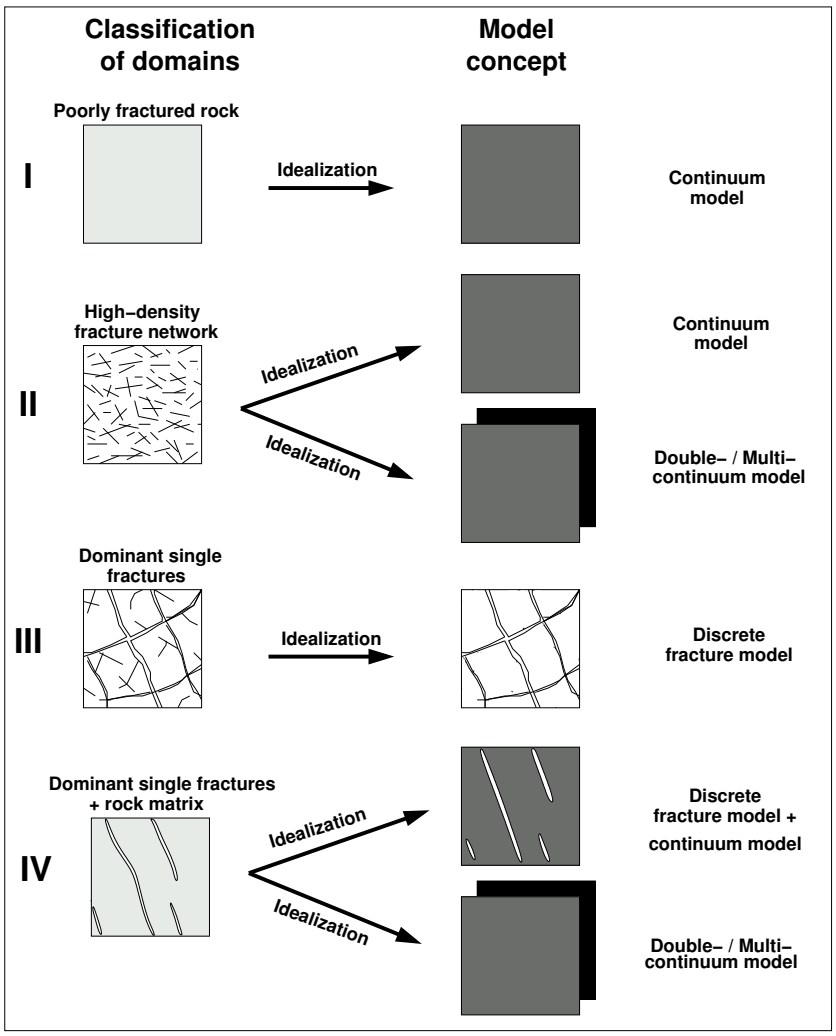


Figure 2.3: Model concepts for the description of fractured porous media (based on Kröhn (1991) and Helmig (1993))

Discrete fracture models [10, 51, 57, 142] allow quantification of many flow and transport phenomena that are not adequately captured by continuum models. An advantage of the discrete fracture approach is that it can account explicitly for the effects of individual fractures on fluid flow and transport. In cases where the matrix is almost impermeable and only interconnected fractures contribute to flow, network models (DFNs) [10, 51] have been used, where the matrix is neglected and the fractures are modeled by line segments in 2D and pipes or planes in 3D. When both fractures and the matrix play a significant role for the flow and transport processes and the model domain cannot be homogenized a single-porosity model concept has been used [57], where both matrix and fractures are explicitly modeled. DFMs are essentially stochastic in that one cannot hope to characterize all fractures accurately [39], and they are best treated in a stochastic framework, by considering Monte Carlo analysis based on multiple realizations of a fracture system. One application of DFN models has has been to generate numerous random DFNs for the purpose of ascribing random effective parameters to subdomains of the rock mass, which would then be treated as a random heterogeneous stochastic continuum, also called stochastic continuum model (SC) [161].

Continuum models may offer advantages over DFMs regarding model size, speed, and modeling of multiphase flow, on the other hand, they may miss local dominant phenomena, regrading network connectivity [56]. A hybrid model (see [10, 29, 51, 103, 119], and references therein) is a usually implemented as a DFM with upscaled matrix properties accounting for some of the fractures. The hybrid model aims at being more computational efficient than the full DFM, and more accurate than the continuum model. Such models can also be used when a fracture system do not possess any homogenization scale and it is difficult to define a REV in the continuum model. They have also been used in cases where site characterization only enables partial delineation of discrete fractures and assigns bulk properties to other regions [29].

In both the continuum and the hybrid models, parameters such as permeability and the transfer function must be determined. Discrete fracture models have been used to determine improved local transfer functions between the matrix and the fractures [92] in dual-porosity models.

2.3 Models for a fracture

An important factor that dictates fracture porosity and permeability is the morphology of the fracture planes [118]. This morphology can be observed observed in core and outcrop and inferred from some well logs. There are four basic types of natural fracture plane morphology [118]:

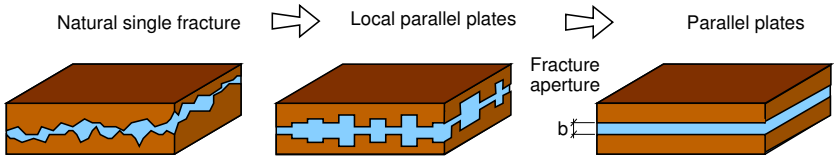


Figure 2.4: Parallel plate concept, after Reichenberger et al. [136]

- open fractures,
- deformed fractures,
- mineral-filled fractures or
- vuggy fractures.

A natural fracture is bounded on both sides by the rock surface. The rough fracture walls do not have an identical profile and the normal tension is carried by contact zones between walls [57].

Fluid flow in a filled fracture can be treated by Darcy's law, since a filled fracture is a porous medium. For open fractures, a different situation arises, which has been addressed by many researchers. A model concept frequently used for open fractures consists of two parallel plates, representing the fracture walls. As illustrated in Figure 2.4, it can be applied locally maintaining a variation in fracture aperture throughout the fracture, or globally, assuming one constant aperture for the total fracture. It is a well-known fact that especially the latter approach is a strong simplification of nature. However, other methods proposed in the literature have not yet found general acceptance [29].

When the parallel-plate concept is applied, it is assumed that the length scale l of the plates is much larger than the distance between them b ($l \gg b$). Furthermore, hydraulically smooth walls and laminar flow are assumed, corresponding to the Poiseuille fluid model. Figure 2.5 shows two parallel plates and the parabola-shaped velocity profile, indicating laminar flow. From the Navier-Stokes equation for laminar single-phase flow of an incompressible Newtonian fluid, it can be shown [25] that the permeability K of the fracture, approximated by the parallel model, is given by

$$K = \frac{b^2}{12}, \tag{2.1}$$

where b is the aperture of the fracture.

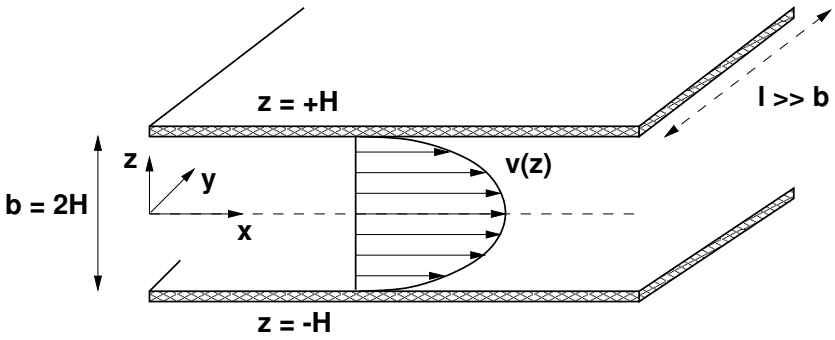


Figure 2.5: Parallel plates, after Reichenberger et al. [136]

Chapter 3

Mathematical model

In the previous chapter we described the development of a conceptual model for a (fractured) porous medium. Having developed a conceptual model for the system, the next step is the representation of the conceptual model in the form of a mathematical model. In principle, a mathematical model can be stated and solved at the microscopic level, as discussed in the previous chapter. However, this approach is not possible, since we cannot describe the geometry and observe and measure quantities at this level. Therefore, the mathematical model is simplified by transforming it from the microscopic level, to a macroscopic, or continuum level, where it is formulated in terms of measurable variables that are averages of microscopic quantities, as described in the previous chapter.

The mathematical models developed in this chapter can be applied to problems including water flow and transport of pollutants in aquifers, flow of hydrocarbons in petroleum reservoirs, and carbon sequestration in geological formations. They consists of sets of partial differential equations which express conservation of mass.

By using local averaging techniques [157] it can be shown that under appropriate assumptions the momentum conservation of the Navier-Stokes equation reduces to Darcy's law on the macroscopic level. The Darcy equation for laminar flow in porous media, developed by Henry Darcy in 1856, gives a relationship between the flow rate and the pressure gradient. It can be written as

$$\boldsymbol{q} = -\frac{\boldsymbol{K}}{\mu}(\nabla \boldsymbol{P} - \rho \boldsymbol{g}), \tag{3.1}$$

where $\boldsymbol{q}$ is the specific discharge vector (commonly denoted the Darcy velocity), $\boldsymbol{K}$ is the permeability tensor μ is the viscosity, ρ is the density, $\boldsymbol{g}$ is the gravity vector, and $\boldsymbol{P}$ is the pressure of the fluid. The permeability tensor is function of the position in the medium, and quantifies the ability of the rock to transmit fluid. The permeability can vary with direction, in which case the porous medium is

called anisotropic, otherwise it is called isotropic.

Since flow only takes place through a part of a cross-sectional area of a porous media, the Darcy velocity $\boldsymbol{q}$ is related to the average (particle) velocity $\boldsymbol{v}$ through

$$\boldsymbol{q} = \phi \boldsymbol{v}, \tag{3.2}$$

as shown by Bear [23]. Here ϕ is the porosity which is a measure of the fraction of void space within an REV. Darcy's law is valid for slow flow of a Newtonian fluid through a porous medium with rigid solid matrix, see [23, 144].

3.1 Single phase flow

The equation governing single phase flow in a porous medium is derived by combining: 1) conservation of mass, 2) Darcy's law, and 3) an equation of state. Consider a typical volume element (control volume), Ω, then conservation of mass implies that the rate of mass accumulated within Ω equals the rate of mass flow across the boundary of Ω plus the amount of mass injected into Ω via wells (sources or sinks). If $\partial\Omega$ is the boundary of Ω with normal vector $\boldsymbol{n}$ and f is the mass flow rate per unit volume injected into Ω, then conservation of mass can be described by

$$\frac{d}{dt}\int_{\Omega} \phi\rho dV = -\int_{\partial\Omega} \rho \boldsymbol{q}\cdot\boldsymbol{n} dS + \int_{\Omega} f\, dV. \tag{3.3}$$

By applying the theorem of Gauss, and using that Equation (3.3) is valid for any volume Ω, we get the parabolic partial differential equation

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla\cdot(\rho\boldsymbol{q}) = f. \tag{3.4}$$

If the fluid is incompressible, i.e., ρ is constant, and if the rock is nondeformable, i.e., ϕ is constant in time, we obtain from Equation (3.4) the elliptic partial differential equation

$$\nabla\cdot\boldsymbol{q} = \frac{f}{\rho}, \tag{3.5}$$

which is also the form of Equation (3.4) for steady state flow.

3.2 Single phase, tracer transport

We now consider the transport of a component, e.g., solute or tracer, contained in a fluid phase. The mass of the tracer per unit volume of fluid is denoted the

concentration, c, of the solute. We consider an ideal tracer that does not affect the fluid's density and viscosity.

Mass conservation for the tracer derived similarly as Equation (3.4) gives,

$$\frac{\partial(\phi c)}{\partial t} + \nabla \cdot \boldsymbol{J} = q_t, \tag{3.6}$$

where $\boldsymbol{J}$ is the total mass flux of the tracer, and q_t is the source term. The total mass flux is composed of advective, dispersive and diffusive fluxes [27], given by

$$\boldsymbol{J} = c\boldsymbol{q} - \boldsymbol{D}\nabla c, \tag{3.7}$$

where $\boldsymbol{D}$ is the tensor of hydrodynamic dispersion. It can be composed of two terms describing molecular diffusion and mechanical dispersion, see e.g. Bear and Verruijt [27].

If dispersion is neglected, and incompressible fluid and porous medium is assumed, Equation (3.6) reduces to

$$\phi \frac{\partial c}{\partial t} + \boldsymbol{q} \cdot \nabla c = q_t. \tag{3.8}$$

3.3 Two phase flow

At last, we consider the flow of two immiscible phases, typically oil and water (o,w), with a single component in a porous media. The extended Darcy's law [82] for each phase α is given by:

$$\boldsymbol{q}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\boldsymbol{K}(\nabla P_\alpha - \rho_\alpha \boldsymbol{g}), \qquad \alpha = o, w \tag{3.9}$$

here $\boldsymbol{q}_\alpha$ are the specific discharge vectors (commonly denoted the Darcy velocities), $\boldsymbol{K}$ is the permeability tensor $k_{r\alpha}$ are the relative permeabilities, μ_α are the viscosities, ρ_α are the densities, $\boldsymbol{g}$ is the gravity vector, and P_α are the phase pressures.

Let S_α denote the fraction of the volume occupied by phase α at a given point in the reservoir. Conservation of mass over a control volume Ω gives,

$$\frac{d}{dt}\int_\Omega (\phi \rho_\alpha S_\alpha) dV + \int_{\partial\Omega} (\rho_\alpha \boldsymbol{q}_\alpha) \cdot \boldsymbol{n} dS = \int_\Omega Q_\alpha. \tag{3.10}$$

By standard arguments we get the local form:

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) + \nabla \cdot (\rho_\alpha \boldsymbol{q}_\alpha) = Q_\alpha, \tag{3.11}$$

where Q_α is the source term for phase α. In addition, we will assume the constitutive relations (equations of state):

$$S_w + S_o = 1 \tag{3.12}$$
$$P_o - P_w = P_c(S_w) \tag{3.13}$$
$$\rho_\alpha = \rho_\alpha(P_w) \tag{3.14}$$
$$\mu_\alpha = \mu_\alpha(P_w) \tag{3.15}$$
$$k_{r\alpha} = k_{r\alpha}(S_w), \tag{3.16}$$

and we assume that the porosity and the permeability are known:

$$\phi = \phi(\boldsymbol{x}, t) \tag{3.17}$$
$$\boldsymbol{K} = \boldsymbol{K}(\boldsymbol{x}, t). \tag{3.18}$$

By inserting the Darcy velocity in (3.11), we obtain the the flow equations for the flow of two phases with a single component in a porous medium:

$$\frac{\partial}{\partial t}(\phi \rho_\alpha S_\alpha) + \nabla \cdot (-\rho_\alpha \frac{k_{r\alpha}}{\mu_\alpha} \boldsymbol{K}(\nabla P_\alpha - \rho_\alpha \boldsymbol{g})) = Q_\alpha, \qquad \alpha = o, w \tag{3.19}$$

This is a system of two nonlinear equations in two unknowns, e.g., P_w and S_w.

We will reformulate the last equations into one pressure equation (elliptic) and one saturation equation (hyperbolic). To obtain the pressure equation, we expand the time derivatives in (3.19), and add the two resulting equations, giving

$$\frac{\partial \phi}{\partial t} + \phi \left(S_w c_w \frac{\partial P_w}{\partial t} + S_o c_o \frac{\partial P_o}{\partial t} \right) + \frac{1}{\rho_w} \nabla \cdot (\rho_w \boldsymbol{q}_w) + \frac{1}{\rho_o} \nabla \cdot (\rho_o \boldsymbol{q}_o) = \frac{Q_w}{\rho_w} + \frac{Q_o}{\rho_o}, \tag{3.20}$$

where we have used (3.12) and

$$c_\alpha = \frac{1}{\rho_\alpha} \frac{\partial \rho_\alpha}{\partial P_\alpha} \tag{3.21}$$

Equation (3.20) is called the pressure equation.

The water velocity can be expressed in terms of the total velocity as

$$\boldsymbol{q}_w = f_w \boldsymbol{q} + h_w \boldsymbol{K}(\nabla P_c + (\rho_w - \rho_o)\boldsymbol{g}), \tag{3.22}$$

where we have used Equation (3.13) and

$$\lambda_\alpha = \frac{k_{r\alpha}}{\mu_\alpha} \tag{3.23}$$
$$h_w(S_w) = \frac{\lambda_o \lambda_w}{\lambda_o + \lambda_w} \tag{3.24}$$
$$f_w(S_w) = \frac{\lambda_w}{\lambda_o + \lambda_w}, \tag{3.25}$$

and the total velocity $\boldsymbol{q}$ is given by

$$\boldsymbol{q} = \boldsymbol{q}_w + \boldsymbol{q}_o. \tag{3.26}$$

By inserting (3.22) into Equation (3.11) we get the saturation equation

$$\frac{\partial}{\partial t}(\phi \rho_w S_w) + \nabla \cdot (\rho_w (f_w \boldsymbol{q} + h_w \boldsymbol{K}(\nabla P_c + (\rho_w - \rho_o)\boldsymbol{g}))) = Q_w \tag{3.27}$$

We will now simplify the the problem by assuming: incompressible fluids, time independent porosity, no capillary pressure, and no gravity. The flow equations (3.20) and (3.27) then simplifies to

$$\nabla \cdot (-\lambda_t \boldsymbol{K} \nabla P) = \frac{Q_w}{\rho_w} + \frac{Q_o}{\rho_o} \tag{3.28}$$

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot (f_w \boldsymbol{q}) = \frac{Q_w}{\rho_w} \tag{3.29}$$

where $\lambda_t = \lambda_w + \lambda_o$.

3.3.1 Streamline formulation

Within streamline simulation, the streamline parameter τ is denoted the time-of-flight (TOF). Since arrival times of particles along streamlines will be important, we relate the time-of-flight to the average particle velocity and not the Darcy velocity. Hence a streamline $\boldsymbol{s}(\tau) = \boldsymbol{x}(\tau)$ is defined by

$$\frac{d\boldsymbol{x}}{d\tau} = \frac{\boldsymbol{q}(\boldsymbol{x}, t)}{\phi}, \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \tag{3.30}$$

We also have:

$$\| \frac{d\boldsymbol{x}}{d\tau} \| = \frac{ds}{d\tau}, \tag{3.31}$$

where $s = \|\boldsymbol{s}\|$. Hence,

$$\frac{ds}{d\tau} = \frac{\|\boldsymbol{q}\|}{\phi}. \tag{3.32}$$

Integrating the last equation gives an expression for the time-of-flight as a function of arc length:

$$\tau(s) = \int_0^s \frac{\phi}{\|\boldsymbol{q}\|} ds'. \tag{3.33}$$

From Equation (3.32), we derive the differential form

$$\|\boldsymbol{q}\|\frac{\partial}{\partial s} = \phi\frac{\partial}{\partial \tau}, \tag{3.34}$$

It follows from (3.28) that $\nabla \cdot \boldsymbol{q} = 0$, away from sources and sinks. In this case the saturation equation (3.29) becomes

$$\phi\frac{\partial S_w}{\partial t} + \boldsymbol{q} \cdot \nabla f_w = 0 \tag{3.35}$$

The last term on the left hand side can be rewritten using the time-of-flight coordinate, using Equation (3.32), as

$$\boldsymbol{q} \cdot \nabla f_w = \|\boldsymbol{q}\|\frac{\partial f_w}{\partial s} = \phi\frac{\partial f_w}{\partial \tau} \tag{3.36}$$

Hence, by using streamlines and time-of-flight, we can simplify the saturation equation (3.35) to a set of 1D hyperbolic equations along streamlines:

$$\frac{\partial S_w}{\partial t} + \frac{\partial f_w(S_w)}{\partial \tau} = 0 \tag{3.37}$$

Note the two different meanings of t and τ in the last equation.

Chapter 4

Numerical model

In the previous chapter, we described simple flow and transport problems in porous media using mathematical models. The description was made up of a partial differential equation, or a system of partial differential equations, together with initial and boundary conditions. In order to solve a given flow problem, this system of equations must be solved for the specific data of the problem. This can be done using analytical methods, or numerical techniques. For most problems of practical interest, because of the irregular shape of the boundaries, the spatial variability of the coefficients appearing in the equations and in the boundary conditions, the nonuniformity of the initial conditions, and the nonanalytic form of the various source and sink terms, analytical solutions are impossible, except for relatively simple problems. Solution of most problems can be obtained only by numerical methods [27].

The main features of the various numerical methods are:

- The solution is sought for the numerical values of the state variables only at specified points in the space and time domains defined for the problem (rather than their continuous variation in these domains).
- The partial differential equations that represent balances of the considered quantities are replaced by a set of algebraic equations written in terms of the sought, discrete values of the state variables at the discrete points in space and time mentioned above.
- The solution is obtained for a specified set of numerical values of various model coefficients (rather than as general relationships in terms of these coefficients).
- Because of the very large number of equations that have to be solved simultaneously, a computer code has to be prepared in order to obtain a solution, using a computer.

Sometimes, the term *numerical model* is used, rather than speaking of a "numerical method of solution" (of the mathematical model). This is justified on the grounds that a number of assumptions are introduced, in addition to those underlying the mathematical model. This makes the numerical model a model in its own right [27].

In this chapter, we will describe three different numerical techniques all related to efficient and accurate simulation of flow and transport in a possibly fractured porous medium.

In the next section, we consider the numerical solution of the porous medium pressure equation. Discrete fracture models, as discussed in Chapter 2, can be formulated in a lower-dimensional framework, where the fractures are modeled in a lower dimension than the matrix, or in an equi-dimensional form, where the fractures and the matrix have the same dimension. When the velocity of the flow field is needed explicitly, as in streamline simulation of advective transport, only the equi-dimensional approach can be used directly. The velocity field for the lower-dimensional model can then be recovered by post-processing which involves expansion of the lower-dimensional fractures to equi-dimensional ones. We compare two different discretization methods for the pressure equation; one vertex-centered approach which can be implemented as either a lower- or an equi-dimensional method, and a cell-centered method using the equi-dimensional formulation.

In Section 4.2 we consider post processing of the velocity field obtained from the pressure solution. For streamline tracing, post-processing is required for a vertex-centered method in general. This is due to the way fluxes are defined. For cell-centered methods fluxes are computed at the boundaries of the original mesh cells, which are typically triangles or quadrilaterals in 2D. For the vertex-centered methods, however, the fluxes are related to the boundaries of control-volumes that coincides with the cells of the dual mesh. These cells are in general polygons in 2D. Since streamline methods requires fluxes on triangles or quadrilaterals, the fluxes of the vertex-centered solution is post-processed using a flux recovery method to obtain new mass conservative flux on a new mesh consisting of quadrilaterals. In addition, when lower dimensional fractures are considered, a fracture expansion must be included in the recovery procedure.

Finally, in Section 4.3 we describe methods for tracing streamlines on irregular grids. A streamline tracing algorithm should satisfy certain tracing quality criteria, e.g., no crossing of streamlines, independence of starting point, reproduction of a uniform flow field, reproduction of symmetry, no premature termination of streamlines in cells or at cell boundaries, good approximation to the true solution, and computational efficiency. We consider grids of quadrilaterals in 2D and of hexahedrons in 3D. The tracing is based on a mapping of an irregular grids cell into a unit square in 2D or a unit cube in 3D. We propose a new method that can reproduce uniform flow in 3D.

4.1 Finite volume methods for elliptic equations

The contrast in permeability between fractures and the porous matrix may span many orders of magnitude and can vary highly in space [57], which is a major difficulty in modeling fluid flow in fractured rock [51]. Due to the complex geometry of the system, unstructured grids are required for the discretization of the domain. Also, the heterogeneous and anisotropic behavior of the permeability is challenging for the numerical modeling of the system, and proper averaging of permeability between the computational cells is crucial [65].

Transport models are used for for understanding and predicting the geological system using results like break-through curves, arrival time and storage behavior. The outcome of transport simulations strongly depends on the distribution of velocities in the fractures [51]. The velocity distribution cannot, in general, be determined analytically, and must be obtained numerically from the solution of a flow (pressure) equation.

Thus, the success of a transport simulation is closely related and linked to the efficiency and accuracy of the numerical scheme for the flow equation. In this thesis, we consider, among others, different numerical schemes for the solution of the flow equation in the presence of fractures. In the following, we present a short literature review over discretization methods for the porous media pressure equation [15, 128] with focus on finite volume methods.

We begin by considering the following equation modeling flow of a single phase in a porous medium,

$$-\nabla \cdot \boldsymbol{K} \nabla P = f, \tag{4.1}$$

where $\boldsymbol{K}$ is the permeability tensor, P is the pressure, and f is the source term. There has been extensive research on developing numerical schemes for Equation (4.1). Desirable properties of the discretization beyond the classical stability and accuracy, include local mass conservation, proper averaging of transmissibilities, discrete maximum principle, and cost-efficiency. These properties are crucial in capturing the important details of the solutions of complicated problems on relatively coarse grids. Methods dealing with discretization of Equation (4.1) can be broadly classified as a) finite difference methods, b) finite element methods, and c) finite volume methods. A further classification is into locally conservative methods [97], and non-conservative methods. Finite volume methods are by definition locally (mass) conservative, whereas the standard Galerkin methods are in general not [159].

Locally conservative finite element methods include the discontinuous Galerkin methods [40], and the mixed finite element methods [13]. Mixed finite elements are attractive due to their accurate approximation of the velocity, and proper treatment of discontinuous coefficients. However, an algebraic system

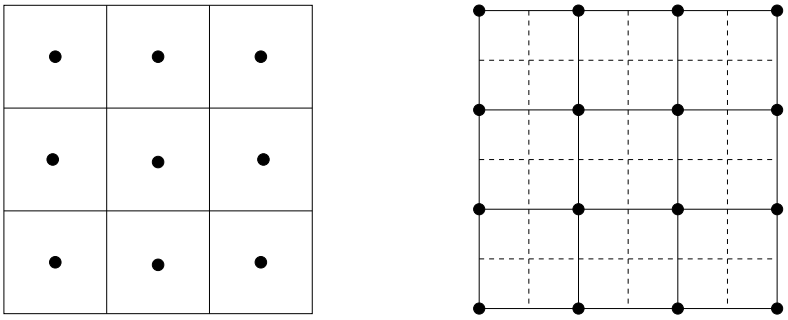


Figure 4.1: Cell-centered and vertex-centered finite volume methods. Left: cell-centered. Right: vertex-centered. Solid lines: primal grid; dashed lines: dual grid; black dots: location of unknowns

of saddle point type [42] must be solved, resulting in a large linear system with a non positive definite matrix. To circumvent these problems, a hybrid form of the MFE method has been used [34]. In this case, the method can be reduced to a symmetric positive definite system for the pressure Lagrange multipliers on the element faces. Advantages of the discontinuous Galerkin method is handling of non-matching grids and an increased order of accuracy. However, the method introduces further degrees of freedom which increase the size of the global matrix.

In the following, we will only consider finite volume methods. Finite volume methods is well suited for simulation of various types of conservation laws [72]. The finite volume method is based on a "balance" approach: a local balance is written for each discretization cell which is often called "control volume"; by the divergence theorem, an integral formulation of the fluxes over the boundary of the control volume is obtained. The fluxes over the boundary are discretized with respect to the discrete unknowns. The most common schemes are either cell-centered (block-centered), in which the primal grid cells are used to define the control volumes, or vertex-centered (point-distributed), in which control volumes are constructed around the primal grid vertices, see Figure 4.1. In this thesis, we consider both vertex-centered and cell-centered approaches. In particular, we will compare a vertex-centered box method with a cell-centered multi-point flux method for a discrete fracture model. The next two sub sections present a short review of the literature on these methods.

4.1.1 Vertex-centered methods

For the vertex-centered methods we will focus on the box methods, also called finite-volume element methods (FVE) [36], control-volume finite element method (CVFEM) [16], generalized difference methods [105], and subdomain collocation

method [87, 163].

The vertex-centered finite volume method has been widely used in computational fluid dynamics, heat transfer, and porous media flow. We consider vertex-centered FVMs that can be formulated as a finite element method, which is characteristic for the box type methods. Generally, a finite element method involves two spaces, the test- and the trial-space. The box method can be cast in the finite element setting with a trial space of piecewise polynomials over the primary grid, and a test space over the dual grid. In the box method, the trial functions will be different from the test functions, hence the box scheme can be treated as a Petrov-Galerkin finite element method [44]. We consider box schemes where the test functions are piecewise constant, and the trial functions are conforming piecewise linear polynomials.

Note that most papers are concerned with theoretical studies of error estimates, due the close connection to the finite element method. The papers often differs in their requirements on the mesh, and in the type of proof used. Papers dealing with applications commonly consider multiphase flow.

For early and non-English literature, we refer to the books of Heinrich [81] and Li et al. [105]. Also Ewing et al. [71] mention some early Russian FVE literature from the 1960ies.

Baliga and Patankar [16, 17] develops the control volume finite-element method. In those papers the method was applied to a convection-diffusion problem. Ramadhyani and Patankar [134] compares the control-volume finite element method with the Galerkin method for the Poisson equation.

For papers on error estimates, see, e.g., [37, 41, 48, 71]. For superconvergence results see [31, 86].

The PhD-thesis of Trujillo [153] investigates high-order FVE methods and also gives a nice introduction to FVE method. Li et al. [105] give a comprehensive overview of generalized difference methods for second order elliptic equations. They describe both first-order and higher-order methods on triangles and quadrilaterals. Rui [138] develops a symmetric FVE method for general self-adjoint elliptic problems. For other types of box schemes, see [54, 112].

There are few applications to single phase flow in the literature, most applications of the vertex-centered approach is to multi-phase flow, see, e.g, [21, 44, 73, 83, 113, 127, 135].

A different type of vertex-centered method is the vertex-centered MPFA-methods, which is different from the box methods presented here, in the sense that rock properties are not constant over the cells of the primal grid, but instead it is constant over the (polygonal) cells (or control-volumes) of the dual grid, see [3, 65]. Since the vertex-centered MPFA methods have very similar formulation as the cell-centered MPFA methods, and papers often discuss both methods in a general framework, we refer to the next section for discussion of the vertex-centered

MPFA method.

4.1.2 Cell-centered methods

In cell-centered methods the control-volumes are associated with the primary grid. This is convenient compared to vertex-centered methods, when considering discontinuous media properties combined with a quadrilateral or triangular primary mesh. In this case it is easy to align the grid edges, and hence, the control-volume boundaries with media discontinuities. This is important considering multi-phase flow, and a reason why many existing petroleum reservoir simulators are based on cell-centered discretizations.

Many types of cell-centered formulations for irregular grids have been proposed in the literature, e.g., integrated finite differences [116], mimetic finite difference methods [146], enhanced finite differences [12], control-volume mixed finite element methods [35], and multi-point flux approximation methods [1].

The integrated finite difference methods has the disadvantage that it is based on the criteria that cell faces are perpendicular to the connections between the cell-midpoints [137], the mimetic finite difference method has many desirable properties like applications to general grids and a symmetric positive definite matrix; however, the method introduces additional cell-edge unknowns. The enhanced finite difference method is very accurate for smooth grids and coefficients, but loses accuracy near discontinuities [156]. The control-volume mixed finite element method solve for both velocity and pressure, similar to the mixed finite element method. Hence, the method can give an accurate approximation of the velocity at cell-edges, but it also introduces additional unknowns compared to methods only solving for the cell-center pressures.

In this review we focus on the multi-point-flux approximation (MPFA) method, also called flux-continuous scheme. Rapid variation in permeability is common for flow in porous media, and for oil reservoirs in particular. For a discrete model, the computational grid and the geology will, in general, interact to produce a full permeability tensor. Early work in the 1970ies and 1980ies assumed either an isotropic permeability tensor or a diagonal tensor such that the discretization could employ a minimal five-point operator, or alternatively, a two-point flux [116, 80]. However, this assumption is only true if the computational grid is aligned with the principal axis of the tensor. The multi-point flux approximation methods were developed as an improvement to the two-point flux approximation when considering irregular grids and a full permeability tensor.

Different research group developed similar methods starting in the beginning of the 1990ies. Most notably among these are the multi-point flux approximations (MPFA) of Aavatsmark et al. [1] and the flux-continuous (FC) schemes of Edwards et al. [65]. The schemes are control-volume distributed, meaning that the

rock properties are assumed constant over control-volumes, and they maintain a single degree of freedom per control-volume. The schemes can be formulated as either vertex-centered or cell-centered [3, 65].

The vertex-centered formulation of the MPFA-methods are different from the box methods discussed in the previous section, in the sense that rock properties are not constant over the cells of the primal grid, but instead it is constant over the (polygonal) cells (or control-volumes) of the dual grid, see [3, 65]. A possible disadvantage of this approach is that it is difficult to align the grid cells with discontinuities in permeability; see Edwards and Pal [64] for an overview of the methods.

A central issue have been if the descretization is symmetric and positive definite (SPD). An SPD formulation is favored theoretically for proving convergence and practically for using simpler and more robust solvers. Methods derived in the physical space is commonly not symmetric, whereas methods derived in the transform space are. However, the latter methods may experience a reduced order of convergence. For papers related to the physical-space based methods, see, e.g, [3, 64, 67, 104, 154], and for papers related to the transform-space based methods, see, e.g., [2, 1, 61, 62, 98, 126].

Different types of MPFA-methods have been developed with different convergence and monotonicity properties. The most popular method has been the O-method. In this method, transmissibility are calculated by requiring continuity of potential and flux at the interfaces of cells in an interaction region. The potential is only required to be continuous at a point of the interfaces, and different locations of the continuity point yield different methods. Let $\eta \in [0, 1]$ be the fraction of the cell half-edge from the edge midpoint to the continuity point and term for each η the resulting method for $O(\eta)$. The $O(\eta)$ method have been considered by, e.g., [60, 61, 65, 126], whereas the specific case of $\eta = 0$, also called just the O-method, is considered by, e.g., [2, 7, 8, 11, 67, 68, 101, 102, 104]. Other methods include the U-method [2, 3, 5, 6], the L-method [9, 38], and the Z-method [123].

Due to the non-variational form of these methods, it has been difficult to obtain analytical error estimates, and numerical convergence rates has been presented in many papers, see, e.g., [6, 7, 68]. However, recently a connection between the mixed-finite element method and the MPFA methods have been established, which have allowed rigorous error estimates to be developed for these methods [11, 96, 98, 156, 160].

Monotonicity issues have been considered by [4, 43, 120, 121, 123]. and the 3D case have been considered by, e.g., [1, 5, 63]. For other MPFA methods, see [43, 147, 154].

4.1.3 The box method

The box methods [18] are vertex-centered FVMs that can be formulated as a finite element method. Generally, a finite element method involves two spaces, the test- and the trial-space. The box method can be cast in the finite element setting with a trial space of piecewise polynomials over the primary grid, and a test space over the dual grid [105]. Since the trial functions will be different from the test functions, the box scheme can be treated as a Petrov-Galerkin finite element method [44].

For the box-scheme considered herein, the test functions are piecewise constant, and the trial functions are conforming piecewise linear polynomials. This kind of box scheme has also been called control-volume finite element method [16], finite volume element method [36], generalized difference method [105], and subdomain collocation method [87, 163].

Consider first the case of no fractures. The domain is discretized with triangular and/or quadrilateral elements E_j, $j = 1, \ldots, N^e$, and vertices $\boldsymbol{\mathcal{V}}_i$, $i = 1, \ldots, N^v$. In the box method, one algebraic equation corresponds to every vertex in the primary mesh. The boxes are usually constructed as a dual mesh of an underlying grid. There are various ways to introduce the dual mesh, e.g., the Voronoi mesh based on the circumcenter [36, 105], or the Donald-mesh, based on the barycenter [18, 44, 71].

For the Donal-mesh the secondary dual mesh is defined by connecting the barycenter (centroid) of each element E_j with edge midpoints. The dual mesh divides each element into three or four sub quadrilaterals depending on the type of element (triangle/quadrilateral). The union of all sub quadrilaterals adjacent adjacent to a vertex $\boldsymbol{\mathcal{V}}_i$ is denoted Ω_i^B, or the control volume associated with $\boldsymbol{\mathcal{V}}_i$, see Figure 4.2 (left).

As for the finite element method, the unknown in Equation (4.1), $P = P(x, y)$ is approximated using basis functions ϕ_i, $i = 1, \ldots, N^v$, weighted by discrete values $\hat{P}_i \equiv P(\boldsymbol{\mathcal{V}}_i)$, i.e., $P \approx \tilde{P} = \sum \phi_i \hat{P}_i$. A basis function is a piecewise continuous function such that its restriction to a triangular element will be a linear function, whereas its restriction to a quadrilateral is an isoparametric mapping of a bilinear function on a reference element [49, 150]. As usual, the basis functions ϕ_i are local functions, meaning that they are identically zero outside Ω_i^B.

The approximation of P with $\tilde{P}$ now leads to a residual in Equation (4.1), $\nabla \cdot \tilde{\boldsymbol{q}} - \nabla \cdot \boldsymbol{q} = \nabla \cdot \tilde{\boldsymbol{q}} - f$, where $\tilde{\boldsymbol{q}} = -\boldsymbol{K} \nabla \tilde{P}$. In the method of weighted residual [163], the weighted integral of the residual over the whole domain is set to zero,

$$\int_\Omega W_i (\nabla \cdot \tilde{\boldsymbol{q}} - f) dV = 0, \qquad i = 1, \ldots, N^v, \tag{4.2}$$

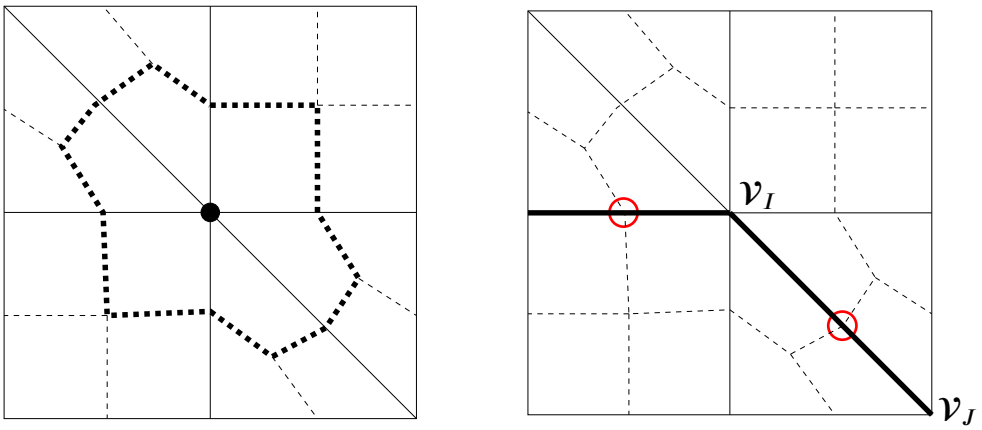


Figure 4.2: Left: Sample grid with no fractures consisting of a primary grid of two quadrilaterals and four triangles. The dual grid is indicated by dashed lines. The control volume associated with the central vertex (black circle) is indicated by the heavy dashed line. Right: Grid with fractures; intersection points indicated with red circles. A fracture intersects the dual mesh at its midpoint. The vertices $\mathcal{V}_I$ and $\mathcal{V}_J$ are the end points of a fracture segment.

for each weight function W_i, $i = 1, \ldots, N^v$. The Box-method uses the simple form,

$$W_i(x) = \begin{cases} 1 & x \in \Omega_i^B \\ 0 & x \notin \Omega_i^B. \end{cases} \tag{4.3}$$

The application of the divergence theorem to (4.2) now leads to a set of N^v surface integrals over the boundaries of each control volume Ω_i^B. For triangular elements, these integrals can be evaluated analytically, whereas for quadrilateral elements the gradient of the pressure is not constant on each element, and numerical integration is required. In the numerical experiments in this paper we used a midpoint rule to evaluate the fluxes for quadrilateral elements.

Assuming an element-wise constant permeability, the N_v integrals can be expressed as

$$\sum_{j=1}^{N_{t,i}} \sum_{k=1}^{3} \gamma_{i,j,k} \hat{P}_{i,j,k} = f_i \qquad i = 1, \ldots, N^v. \tag{4.4}$$

Here, we have assumed a grid of only triangular elements, and the sum on k is running over the three vertices of each triangle j contributing to the control volume Ω_i^B. There are $N_{t,i}$ triangles contributing to Ω_i^B, and $\hat{P}_{i,j,k}$ refers to the pressure at local corner number k of triangle j. Furthermore, f_i is the integral of the source

term f over Ω_i^B, and $\gamma_{i,j,k}$ can be expressed as

$$\gamma_{i,j,k} = \left[\frac{\boldsymbol{K}_{i,j}}{2T_{i,j}}\boldsymbol{\nu}_{i,j,k}\right] \cdot (\boldsymbol{n}_{i,j}^1 + \boldsymbol{n}_{i,j}^2). \tag{4.5}$$

Here $\boldsymbol{n}_{i,j}^1$ and $\boldsymbol{n}_{i,j}^2$ are the two outward normal vectors (having length equal to the length of the interface) associated with the part of $\partial\Omega_i^B$ lying within triangle j, $\boldsymbol{K}_{i,j}$ and $T_{i,j}$ are the permeability tensor and the area of triangle j, respectively. The vectors $\boldsymbol{\nu}_{i,j,k}$ and the area $T_{i,j}$ results from taking the gradient of the linear pressure variation in triangle j as shown in, e.g., [6].

The last part of the box method consists of the assembly of the coefficient matrix and the right hand side of the linear system for the discrete (vertex) pressures giving N^v linear equations for the N^v unknown vertex pressures $\hat{P}_i$. Note that, for Dirichlét boundary conditions, the number of Dirichlét boundary nodes must be subtracted from N^v, see below for a discussion of boundary conditions.

The linear system can be written in matrix form as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{4.6}$$

where $\boldsymbol{A}$ is the $N^v \times N^v$ coefficient matrix, $\boldsymbol{u}$ is an $N^v \times 1$ vector of the unknown vertex pressures, and $\boldsymbol{b}$ is an $N^v \times 1$ vector representing the source terms or boundary conditions. Since the linear system is symmetric positive definite, it can be solved rapidly using, e.g., the preconditioned conjugate gradient method.

Discrete fractures

In n-dimensional space, fractures are often modeled as $(n-1)$-dimensional objects [57, 135]. In 2D, this means that fractures are associated with the edges of the finite element mesh. Hence, edges are either matrix edges (edges containing no fractures), or fracture edges (edges containing a fracture), see Figure 4.2 (right). The fracture edges are denoted by $\mathcal{F}_k$, $k = 1, \ldots, N_f$, where N_f is the number of fracture edges. A fracture edge intersects a control volume boundary of the dual mesh at its midpoint, see Figure 4.2 (right).

Each fracture edge $\mathcal{F}_k$ will also have an associated aperture a_k, permeability k_k, and length l_k. Refer to Chapter 2.3 for a discussion of how to determine the permeability for open fractures.

The length l_k is the length of the fracture edge, the permeability k_k is assumed to be isotropic such that a two-point flux can be applied inside the fracture and the aperture a_k is not resolved explicitly in the grid, but used for computing fracture fluxes, as discussed next.

Consider a fracture edge $\mathcal{F}_k$ with end points at nodes $\mathcal{V}_I$, $1 \leq I \leq N^v$, and $\mathcal{V}_J$, $1 \leq J \leq N^v$, as in Figure 4.2 (right). Then the flux $Q_{i,j,k}$ out of Ω_I^B at the

midpoint of $\mathcal{F}_k$ is approximated as a two-point flux

$$Q_k = \frac{k_k a_k}{l_k}(\hat{P}_I - \hat{P}_J). \tag{4.7}$$

The above fluxes are then taken into account when evaluating the surface integrals resulting from Equation (4.2), and when assembling the linear system (4.6).

Boundary conditions

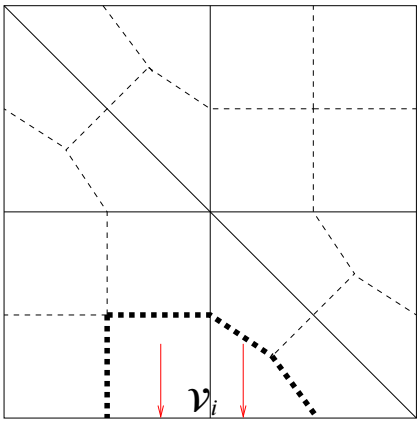


Figure 4.3: A boundary control volume is indicated by the heavy dashed line.

Mass conservation for interior control volumes, i.e. those not having common boundary with $\partial\Omega$, is always enforced through Equation (4.4). A control-volume next to the boundary $\partial\Omega$ can be associated with either a Neumann or a Dirichlét condition. Mass conservation is enforced for Neumann control-volumes. In order to keep the number of equations equal to the number of unknowns, mass conservation can, however, not be enforced for Dirichlét control volumes. Consider the node $\mathcal{V}_i$ in Figure 4.3. Assume Ω_i^B is a Neumann control-volume, then there will be given fluxes (the boundary conditions) g_i^L and g_i^R as indicated with the red arrows in the figure. Let $g_i = g_i^L + g_i^R$, then mass conservation for Ω_i^B is obtained by adding g_i to the left hand side of Equation (4.4).

For interior control volumes, where Equation (4.4) involves some pressures on a Dirichlét boundary, the exact pressure is inserted and the resulting terms are moved to the right hand side.

4.1.4 The MPFA O-method

The lower-dimensional treatment of fractures in the box methods has been popular since it allows for easy mesh generation and a well-conditioned linear systems.

However, streamlines cannot be traced directly for this approach, since there is no transversal flow information in the lower-dimensional fractures (see also the discussion of flux-recovery in the next section). It is therefore of interest to consider equi-dimensional methods. In this section we consider the cell-centered MPFA method for equi-dimensional fractures. Note that the MPFA method also has been used with lower-dimensional fractures [85], however this approach can only be used for low-permeable fractures.

In cell-centered finite volume methods the control-volumes are associated with the primary grid. This is convenient compared to vertex-centered methods, when considering discontinuous media properties combined with a quadrilateral or triangular primary mesh. In this case it is easy to align the grid edges, and hence, the control-volume boundaries with media discontinuities.

Here we use a multi-point flux O-method (MPFA O-method), briefly described below. For an overview of different MPFA schemes see, e.g., [3, 1, 6, 64, 154], and references therein. The letter "O" comes from the shape of the polylines connecting the involved grid points in a cell-stensil [3]. The MPFA O-method is based on full flux continuity across cell edges, continuity of the pressure at the midpoint of the cell edges, and mass conservation for each cell. The pressures at cell edges is eliminated locally using the flux continuity constraints, and the global system is expressed in terms of the cell center pressures.

For the MPFA O-method the control volumes are associated with the original mesh, whereas for the box method, control volumes were associated with the dual mesh. By integrating Equation (4.1) over a control volume Ω_i^M and applying Gauss' theorem we have

$$-\int_{\partial\Omega_i^M} \boldsymbol{K}\nabla P\cdot\boldsymbol{n}dS = \int_{\Omega_i^M} f\,dV, \qquad i=1,\ldots,N^e, \tag{4.8}$$

where $\boldsymbol{n}$ is the unit outward normal vector to $\partial\Omega_i^M$. The only unknown in the last equation is the pressure, which will be approximated at the center of each cell (cell-centered method), in contrast to the Box-method, where the pressure is approximated at the vertices of the grid (vertex-centered method). Hence, Equation (4.8) gives us N^e equations to determine the N^e unknown cell center pressures. By assuming a constant permeability $\boldsymbol{K}_i$ for each control volume, we can write the integral on the left hand side of Equation (4.8) as

$$\int_{\partial\Omega_i^M} -\boldsymbol{K}_i\nabla p\cdot\boldsymbol{n}dS = \sum_{j=1}^{N_{e,i}}\int_{\Gamma_{i,j}} -\boldsymbol{K}_i\nabla p\cdot\boldsymbol{n}_{i,j}dS = \sum_{j=1}^{N_{e,i}} Q_{i,j},\ i=1,\ldots,N^e, \tag{4.9}$$

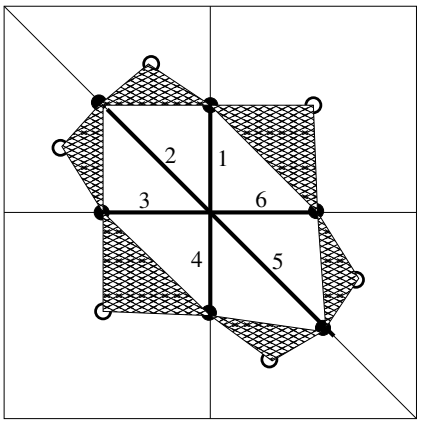


Figure 4.4: An interaction region, variational triangles, half-edges and pressure nodes, respectively, for the central vertex of a simple grid. Variational triangles are cross hatched; the half-edges are the heavy line segments numbered 1-6; solid circles indicate auxiliary (temporary) pressure nodes, and non-solid circles indicate cell center pressures.

where $N_{e,i}$ is the number of boundary edges of Ω_i^M, i.e., $N_{e,i}$=3 for a triangle and $N_{e,i}$=4 for a quadrilateral, $\Gamma_{i,j}$ is boundary edge number j of Ω_i^M, $\boldsymbol{n}_{i,j}$ is the corresponding outward unit normal vector, and $Q_{i,j}$ is defined by Equation (4.9). In order to determine the fluxes $Q_{i,j}$ in Equation (4.9) in terms of the cell center pressures, we use a multi-point flux approximation (MPFA).

The MPFA method is based on dividing the grid into so-called interaction regions $\mathcal{I}_i, i = 1, \ldots, N^v$, associated with each vertex $\mathcal{V}_i$ of the grid. The interaction regions are equal to the control-volumes for the box method, see the previous section. For each interaction region $\mathcal{I}_i$ there is a set of half-edges $E_{i,j}$, see Figure 4.4. On each side of a half-edge there will be two grid cells, denoted "L" and "R". Furthermore, each interaction region also contains a set of variational triangles $\mathcal{T}_{i,j}^{\alpha}$, $\alpha = L, R$, one for each cell α of half-edge $E_{i,j}$, see Figure 4.4.

Within each $\mathcal{T}_{i,j}^{\alpha}$, a linear pressure variation is assumed. This is done by fixing the pressure at the corners of each $\mathcal{T}_{i,j}^{\alpha}$. Thus, in addition to the cell center pressures, we introduce (temporary) pressures at the points indicated with the solid circles in Figure 4.4. The linear pressure in each $\mathcal{T}_{i,j}^{\alpha}$ is now assumed to be valid also for calculating the flow across each of the two half-edges within $\mathcal{I}_i$ that are closest to $\mathcal{T}_{i,j}^{\alpha}$, i.e., those with a common point with $\mathcal{T}_{i,j}^{\alpha}$.

Hence, the pressure gradient will be constant on each side of a half-edge. From Equation (4.8) we see that the flow rate $\boldsymbol{q}$ must be given as

$$\boldsymbol{q} = -\boldsymbol{K}\nabla P. \tag{4.10}$$

Substituting the constant pressure gradient into the last equation, we can express

the flux across each half-edge $E_{i,j}$ as

$$q_{i,j}^{\alpha} = \sum_{k=1}^{3} t_{i,j,k}^{\alpha} P(\boldsymbol{x}_{i,j,k}^{\alpha}), \tag{4.11}$$

where

$$t_{i,j,k}^{\alpha} = \frac{|E_{i,j}|}{2T_{i,j}^{\alpha}} \left(\boldsymbol{K}_{i,j}^{\alpha} \boldsymbol{\nu}_{i,j,k}^{\alpha} \right) \cdot \boldsymbol{n}_{i,j}, \tag{4.12}$$

and $P(\boldsymbol{x}_{i,j,k}^{\alpha})$ is the pressure at corner k of $\mathcal{T}_{i,j}^{\alpha}$. The sign of the unit normal vector $\boldsymbol{n}_{i,j}$ to $E_{i,j}$ is not important, but the same sign must be chosen for each value of α. In the last equation, $|E_{i,j}|$ is the length of $E_{i,j}$; $T_{i,j}^{\alpha}$ and $\boldsymbol{\nu}_{i,j,k}^{\alpha}$ are the area and normal vectors of $\mathcal{T}_{i,j}^{\alpha}$, resulting from taking the gradient of a the linear pressure, see, e.g., [6].

Flux continuity across edge $E_{i,j}$ can now be expressed as $q_{i,j}^{L} = q_{i,j}^{R}$, or

$$A_i \boldsymbol{u}_i + B_i \boldsymbol{v}_i = C_i \boldsymbol{u}_i + D_i \boldsymbol{v}_i, \qquad i = 1, \ldots, N^e, \tag{4.13}$$

where $\boldsymbol{u}_i$ is a vector of cell center pressures, and $\boldsymbol{v}_i$ contains the auxiliary pressures (which we want to eliminate), and the matrices A_i, B_i, C_i, and D_i, contains the t-coefficients.

Note that since the pressure is required to be continuous at the auxiliary pressure nodes, i.e., at the solid circles in Figure 4.4, there will be only one unknown at these points, and hence each $\boldsymbol{v}_i$ has exactly N_i^E components, where N_i^E is the number of half-edges within $\mathcal{I}_i$. It can also be shown that the matrices B_i and D_i are square $N_i^E \times N_i^E$ matrices.

We here assume that the interaction region $\mathcal{I}_i$ correspond to an interior corner; for boundary corners and implementation of boundary conditions, see the next section. For interior corners, the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ will also contain the same number (N_i^E) of unknowns, such that A_i and C_i are also square $N_i^E \times N_i^E$ matrices.

Due to the continuity condition (4.13) the flux $q_{i,j}$ across edge $E_{i,j}$ is well defined as either $q_{i,j}^{L}$ or $q_{i,j}^{R}$, and can be expressed as, e.g., $q_{i,j} = q_{i,j}^{L}$, which is expressed in matrix form as

$$\boldsymbol{q}_i = A_i \boldsymbol{u}_i + B_i \boldsymbol{v}_i, \tag{4.14}$$

where the j-th component $\boldsymbol{q}_i$ is $q_{i,j}$. Finally, we eliminate $\boldsymbol{v}_i$ by using (4.13),

$$\boldsymbol{q}_i = (A_i + B_i(B_i - D_i)^{-1}(C_i - A_i))\boldsymbol{u}_i = T_i \boldsymbol{u}_i \tag{4.15}$$

Since the components of $\boldsymbol{q}_i$ represents fluxes over half edges adjacent to a given corner $\mathcal{V}_i$, the flux over edge $\Gamma_{k,j}$ of grid cell Ω_k^M (see Equation (4.9)) can be

found be combining certain components $\boldsymbol{q}_i$, $i = A, B$, where $\boldsymbol{\mathcal{V}}_A$ and $\boldsymbol{\mathcal{V}}_B$ are the end points of $\Gamma_{k,j}$.

In this way, we assemble expressions for each of the $Q_{i,j}$ in Equation (4.9) and inserting these into (4.8), we obtain N^e linear equations that can be solved for the N^e unknown cell center pressures. The linear system is expressed as

$$A\boldsymbol{u} = \boldsymbol{b}, \tag{4.16}$$

where A is the $N^e \times N^e$ coefficient matrix, $\boldsymbol{u}$ is an $N^e \times 1$ vector of the unknown cell center pressures, and $\boldsymbol{b}$ is an $N^e \times 1$ vector representing the source terms for each Ω_k^M, $k = 1, \ldots, N^e$, see Equation (4.8).

Boundary conditions

Dirichlét boundary conditions

Dirichlét boundary conditions have been investigated in [66, 68]. First, the easiest case is when the pressure is known at the cell cent res of the cells closest to the boundary. The pressure in cell Ω_k^M can be fixed to, e.g. u_k, by redefining row k of A in (4.16) to be $\boldsymbol{e}_k$, where $\boldsymbol{e}_k$ is an $1 \times N^e$ vector with zeros everywhere except for position k, where it is one, and then redefining component k of $\boldsymbol{b}$ to be u_k.

Second, if the pressure are known at the midpoints of boundary edges, the auxiliary pressures $\boldsymbol{v}_i$, see Figure 4.4 (right), can be used. For a boundary edge, either $q_{i,j,1}$ or $q_{i,j,2}$ in Equation (4.13) does not exist. We can choose to define the k-index in $q_{i,j,k}$ such that $q_{i,j,1}$ is always defined. Then Equation (4.14) is also always defined, even for boundary corners. However, Equation (4.13) is not well defined for a boundary corner. We rewrite this system for a boundary corner as

$$\tilde{A}_i\boldsymbol{u}_i + \tilde{B}_i\boldsymbol{v}_i = \tilde{C}_i\boldsymbol{u}_i + \tilde{D}_i\boldsymbol{v}_i + \boldsymbol{\gamma}_i, \qquad i\text{=boundary corners} \tag{4.17}$$

where $\tilde{A}_i = A_i$, $\tilde{B}_i = B_i$, $\tilde{C}_i = C_i$, and $\tilde{D}_i = D_i$, for rows corresponding to non-boundary edges. For rows corresponding to boundary edges, the rows of $\tilde{A}_i$, $\tilde{C}_i$, and $\tilde{D}_i$, are set to zero, whereas the rows of $\tilde{B}_i$ is set to zero, except for the diagonal element, which is set to one. The $N_i^{\mathcal{E}} \times 1$-vector $\boldsymbol{\gamma}_i$ is zero everywhere, except for components corresponding to boundary edges, where it is equal to the fixed pressure at this edge.

Now $\boldsymbol{v}_i$ can be expressed as

$$\boldsymbol{v}_i = (\tilde{B}_i - \tilde{D}_i)^{-1}(\tilde{C}_i - \tilde{A}_i)\boldsymbol{u}_i + (\tilde{B}_i - \tilde{D}_i)^{-1}\boldsymbol{\gamma}_i, \tag{4.18}$$

which is inserted in (4.14), such that the fluxes can be expressed in terms of the cell center pressures.

Neumann boundary conditions

At a boundary edge $\mathcal{E}_{i,j_1}$, the flux continuity condition (4.13) is rewritten as

$$q_{i,j_1,1} = \beta_{i,j_1}, \tag{4.19}$$

where β_{i,j_1} is a specified flux at the boundary edge. For convenience, we will also set $\beta_{i,j} = 0$ for non-boundary edges $\mathcal{E}_{i,j}$. Then Equation (4.13), is rewritten as

$$A_i \boldsymbol{u}_i + B_i \boldsymbol{v}_i = \hat{C}_i \boldsymbol{u}_i + \hat{D}_i \boldsymbol{v}_i + \boldsymbol{\beta}_i, \tag{4.20}$$

where the j-th component of the vector $\boldsymbol{\beta}_i$ is $\beta_{i,j}$, and $\hat{C}_i = C_i$, and $\hat{D}_i = D_i$, for non-boundary edges. For boundary edges, we have set the corresponding row in $\hat{C}_i$ and $\hat{D}_i$ to zero. We can now eliminate $\boldsymbol{v}_i$ as in the previous paragraphs.

Some corners may have combined Dirichlét and Neumann boundary edges. The procedures above are then combined.

4.2 Flux recovery

The precision of streamline tracing strongly depends on the accuracy of the velocity field [109]. Vertex-centered methods for flow simulation, like the box method, compute mass conservative discrete fluxes at polygonal control-volume boundaries. However, streamline tracing requires fluxes at quadrilateral or triangular cells. Computing fluxes at the primary mesh by differencing the finite element representation of the solution, is not an option since it results in discontinuous fluxes at element boundaries [149].

On the basis of work of *Cordes and Kinzelbach* [52] and *Prévost* [133], a flux recovery for a two-dimensional fracture-matrix system is used to obtain continuous fluxes on a sub quadrilateral grid. Additionally, lower-dimensional fractures, which are assumed in the flow simulation, have to be extended to equidimensional ones to obtain well-defined velocities in the fractures, see [75]. Fractures are expanded such that the resulting 2D fractures have a width equal to the associated fracture aperture d and the 1D fracture is the center line.

Figure 4.5 shows the five cases that are most likely to occur in a discretized fracture-matrix system. The control volumes are classified as: (type 1) no fractures, (type 2) a single crossing fracture, (type 3) two crossing fractures, (type 4) a single ending fracture, and (type 5) an ending fracture and one passing through. The flux recovery for control volumes is presented here in detail for type 2, and is briefly discussed for the other types afterwards. The expansion procedure for the general case is discussed in [75].

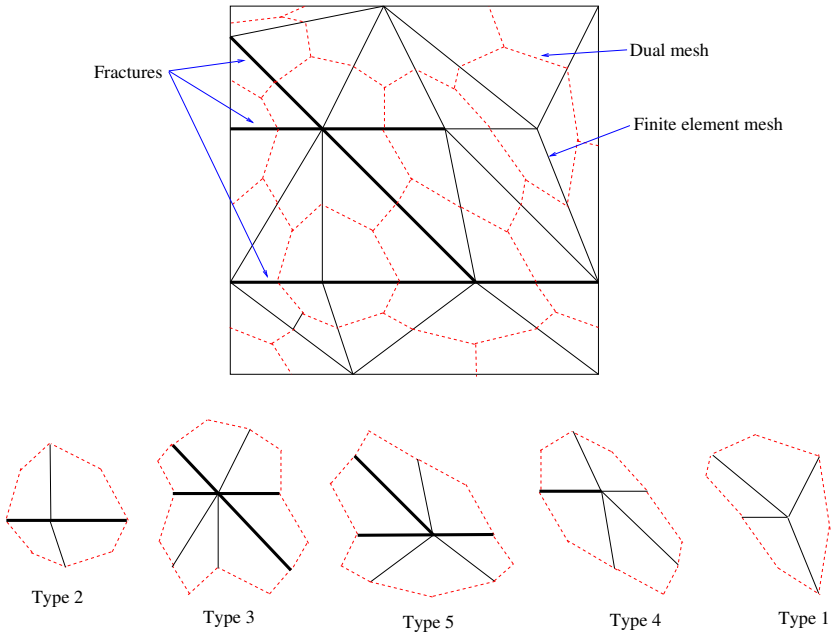
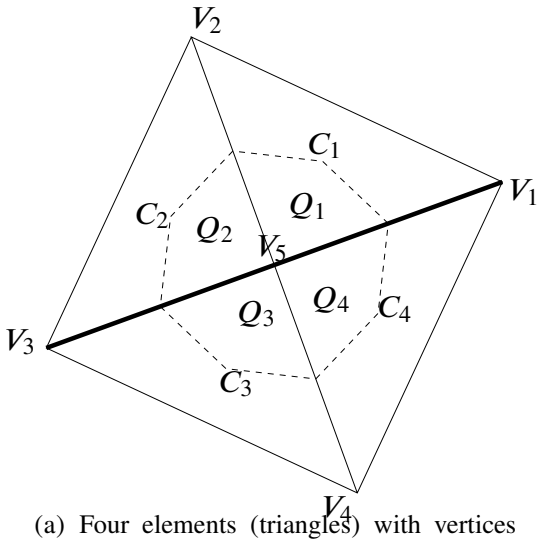


Figure 4.5: Top: Triangular finite element mesh with fractures. Fractures are indicated with heavy lines. Bottom: The different types of control-volumes considered.

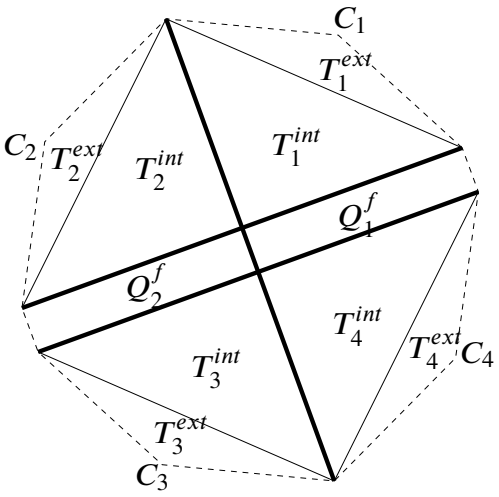
4.2.1 Control volumes with single fractures

A simple prototype control volume of type 2 is shown in Figure 4.6(a). The control volume Ω_{cv} of the central vertex V_5, indicated with the dashed line, consists of the union of N quadrilateral sub control-volumes denoted Q_j, where $j = 1, \ldots, N$, (in this simple case $N = 4$), see Figure 4.6(a). We order Q_j counter clock-wise such that the common edge between Q_1 and Q_N coincides with the fracture. Next, the 1D fracture is expanded to a 2D fracture with aperture d, such that two new rectangular sub control-volumes Q_1^f and Q_2^f are created inside the control volume, see Figure 4.6(b). Each quadrilateral Q_j is split into two triangles, one interior T_j^{int} and one exterior T_j^{ext}, relative to the central node of the control volume, see also Figure 4.6(b). Only the part of the control volume composed of the interior triangles T_j^{int} and the fracture sub control-volumes Q_1^f and Q_2^f is considered in the flux recovery, see Figure 4.6(c).

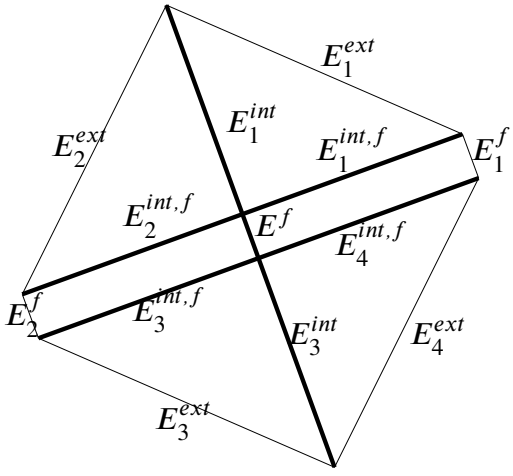
From the flow simulation, fluxes are given over the exterior faces of the control volume Ω_{cv}, as indicated by the dashed line segments in Figure 4.6(b). Hence the fluxes over the exterior edges E_j^{ext} in Figure 4.6(c) are known by mass conservation. The recovery procedure now proceeds by calculating additional conservative fluxes on all remaining interior matrix edges, i.e., the edges (E_j^{int} and $E_j^{int,f}$) and



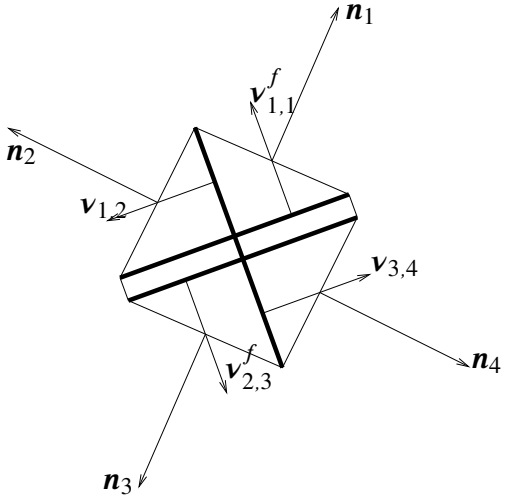
(a) Four elements (triangles) with vertices $V_1, \ldots, V_5$. A control-volume (dashed line) is associated with V_5.



(b) The control-volume in the left figure. The fracture has been expanded to a 2D element with aperture d.



(c) Control-volume edge numbering



(d) Control-volume normal vectors

Figure 4.6: Flux recovery for a control volume with an internal fracture

the interior fracture edge (E_f), see Figure 4.6(c). The fluxes are obtained indirectly by computing a constant Darcy velocity $\boldsymbol{q}_j$ for each interior triangle T_j^{int}.

The constant Darcy velocities $\boldsymbol{q}_j$ and the fracture interior fluxes F_f must satisfy the following conditions:

- Flux continuity at the exterior edges T_j^{ext}, i.e.,

$$\boldsymbol{q}_j \cdot \boldsymbol{n}_j = F_j, \qquad j = 1, \ldots, N, \tag{4.21}$$

  where $\boldsymbol{n}_j$ is the outward normal vector to E_j^{ext} as shown in Figure 4.6(d), with length equal to the length of E_j^{ext}, and F_j are the known fluxes at the corresponding edges.

- Flux continuity over common interior edges of the triangles T_j^{int}:

$$\boldsymbol{q}_j \cdot \boldsymbol{\nu}_{j,j+1} = \boldsymbol{q}_{j+1} \cdot \boldsymbol{\nu}_{j,j+1}, \qquad j = 1, \ldots, M-1, M+1, \ldots, N-1, \tag{4.22}$$

  where M is the number such that T_M^{int} and T_{M+1}^{int} have edges in common with Q_2^f, see Figure 4.6(b). In our simple case, $M = 2$. Furthermore, $\boldsymbol{\nu}_{j,j+1}$ is the normal vector to E_j^{int} pointing from T_j^{int} to T_{j+1}^{int} and has length equal to E_j^{int}.

- Mass conservation for one fracture quadrilateral, e.g., Q_1^f:

$$\boldsymbol{q}_1 \cdot \boldsymbol{\nu}_{1,1}^f - \boldsymbol{q}_N \cdot \boldsymbol{\nu}_{1,1}^f + F^f + F_1^f = 0, \tag{4.23}$$

  where F^f is the unknown flux over the fracture interior edge E^f, F_1^f is the given flux over the edge of the expanded fracture quadrilateral Q_1^f, see Figure 4.6(c). The normal vector $\boldsymbol{\nu}_{1,1}^f$ is shown in Figure 4.6(d). Note that we consider only mass conservation for one of the fracture quadrilaterals; if all of Equations (4.21), (4.22), and (4.23) hold, mass conservation for the other fracture quadrilateral is automatically fulfilled since the total fluxes out of Ω_{cv} is mass conservative, i.e., is equal to zero.

A system of $2N - 1$ linear equations has now been set up. However, a total number of $2N + 1$ ($2N$ from the $\boldsymbol{q}_j$ and one from the flux F^f) unknown components must be determined. The remaining two conditions needed to close the system are obtained by requiring the gradient of the pressure field to be irrotational [52]. Assuming the velocity field to be given by Darcy's law

$$\boldsymbol{q} = -\frac{\boldsymbol{K}}{\mu} \nabla p, \tag{4.24}$$

and rearranging and taking the curl gives

$$\nabla \times \mu \boldsymbol{K}^{-1} \boldsymbol{q} = -\nabla \times \nabla p = 0, \tag{4.25}$$

since the curl of a gradient is always zero. Assuming the viscosity μ to be constant, we have from the theorem of Stokes,

$$\int_{\Omega} \nabla \times \boldsymbol{K}^{-1} \boldsymbol{q} dV = \oint_{\Gamma} \boldsymbol{K}^{-1} \boldsymbol{q} \cdot d\boldsymbol{s} = 0. \tag{4.26}$$

Consider the reduced control volume in Figure 4.6(c). The fracture separates it into two regions Ω_1 and Ω_2. The domain Ω_1 contains the triangles T_j^{int}, $j = 1, \ldots, M$, and Ω_2 contains T_j^{int}, $j = M+1, \ldots, N$. The boundary of Ω_i is denoted Γ_i, $i = 1, 2$. From Equation (4.26), we then have $\oint_{\Gamma_i} \boldsymbol{K}^{-1} \boldsymbol{q} \cdot d\boldsymbol{s} = 0$, $i = 1, 2$. The part of these line integrals taken along the boundary between the fracture and the matrix, are to be evaluated within the fracture. Hence, the Integral equation (4.26) over Γ_1 can expressed as

$$\sum_{j=1}^{M} \int_{E_j^{ext}} \boldsymbol{K}_j^{-1} \boldsymbol{q}_j \cdot d\boldsymbol{s} + \int_{E_2^{int,f}} k_F^{-1} \boldsymbol{q}_2^f \cdot d\boldsymbol{s} + \int_{E_1^{int,f}} k_F^{-1} \boldsymbol{q}_1^f \cdot d\boldsymbol{s} = 0, \tag{4.27}$$

where $\boldsymbol{K}_j$ denote the constant permeability in T_j^{int}. The permeability in the fracture is assumed to be a scalar, denoted k_F. The velocities $\boldsymbol{q}_i^f$, $i = 1, 2$, in the fracture quadrilaterals Q_i^f, are generally not constant, but assumed to be given by linear interpolation of the edge fluxes using Pollock's method [131].

In the first two integrals in (4.27), both $\boldsymbol{K}_j$ and $\boldsymbol{q}_j$ are constant, $j = 1, \ldots, M$. Hence,

$$\sum_{j=1}^{M} \int_{E_j^{ext}} \boldsymbol{K}_j^{-1} \boldsymbol{q}_j \cdot d\boldsymbol{s} = \sum_{j=1}^{M} \boldsymbol{K}_j^{-1} \boldsymbol{q}_j \cdot \boldsymbol{t}_j = \sum_{j=1}^{M} \boldsymbol{K}_j^{-T} \boldsymbol{t}_j \cdot \boldsymbol{q}_j, \tag{4.28}$$

where the tangent vectors $\boldsymbol{t}_j$ corresponds to a 90 degrees counter-clockwise rotation of $\boldsymbol{n}_j$.

Using Pollock's method for the velocity $\boldsymbol{q}_i^f$, $i = 1, 2$, in the fracture quadrilaterals, it can be shown that the last two integrals in (4.27) can be written,

$$\int_{E_2^{int,f}} k_F^{-1} \boldsymbol{q}_2^f \cdot d\boldsymbol{s} = \frac{(-F_2^f - F^f)|E_2^{int,f}|}{2k_F|E_2^f|} \tag{4.29}$$

$$\int_{E_1^{int,f}} k_F^{-1} \boldsymbol{q}_1^f \cdot d\boldsymbol{s} = \frac{(F_1^f - F^f)|E_1^{int,f}|}{2k_F|E_1^f|}, \tag{4.30}$$

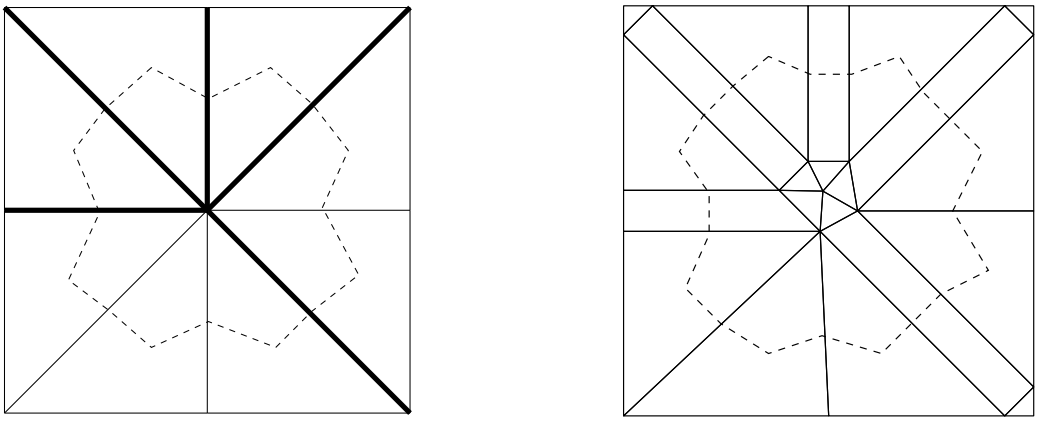


Figure 4.7: Five fractures meeting. Left: Unexpanded fractures. Right: Expanded fractures.

where $|\cdot|$ denotes length of edge, e.g., $|E_1^f|$ is the length of edge E_1^f. Note that the fracture flux F^f over the edge E^f is positive for flow out of Q_1^f, and the fluxes F_1^f and F_2^f are positive for flow out of the corresponding fracture quadrilateral.

As a result Equation (4.27) can be written

$$\sum_{j=1}^{M} \boldsymbol{K}_j^{-T} \boldsymbol{t}_j \cdot \boldsymbol{q}_j + \frac{(-F_2^f - F^f)|E_2^{int,f}|}{2k_F|E_2^f|} + \frac{(F_1^f - F^f)|E_1^{int,f}|}{2k_F|E_1^f|} = 0. \tag{4.31}$$

A similar similar argument can be used to show that the zero curl condition for Ω_2 can be written

$$\sum_{j=M+1}^{N} \boldsymbol{K}_j^{-T} \boldsymbol{t}_j \cdot \boldsymbol{q}_j - \frac{(-F_2^f - F^f)|E_2^{int,f}|}{2k_F|E_2^f|} - \frac{(F_1^f - F^f)|E_1^{int,f}|}{2k_F|E_1^f|} = 0. \tag{4.32}$$

Equations (4.21), (4.22), (4.23), (4.31), and (4.32) now constitutes a set of linear equations that can be solved for the unknown velocities $\boldsymbol{q}_j$ and the fracture flux F^f.

The general case of a fracture-matrix control volume with ϱ fractures meeting at a vertex is now described briefly. A new mesh of expanded fractures is constructed by introducing a polygon $\mathcal{M}$ with ϱ edges at the overlapping area of the ϱ expanded fractures. The ϱ fractures now become ϱ trapezoidal elements and the central polygon $\mathcal{M}$ is divided into ϱ triangles, each having one vertex at the centroid of the polygon. A sketch of a case for $\varrho = 5$ is shown in Figure 4.7. The flux recovery method for two fractures described previously can now be extended to the case of ϱ fractures. The exterior flux continuity conditions in Equation (4.21)

remain the same. However, the interior continuity equations in Equation (4.22) are reduced from $N-2$ to $N-\varrho$ due to the presence of more fractures within the control volume. Furthermore, there are now ϱ fractures for which Equation (4.23) has to be imposed. Note that we do not impose mass conservation for the central polygon $\mathcal{M}$ for the same reason as discussed after Equation (4.23). These increased number of constraints are counterbalanced by an increased number of unknown fracture fluxes. Next, Equations (4.31) and (4.32) have to be extended to ϱ curl relations, instead of two. Finally, after the system of equations have been solved, the computed solution for the ϱ fracture fluxes are used to compute the interior fluxes for a triangulation of the central polygon $\mathcal{M}$ by following the original method of Cordes and Kinzelbach [52]. For other types of control volumes, a similar concept of fracture expansion and flux recovery is applied.

We do not consider the 3D case, however extensions of the flux recovery procedure to the 3D case without fractures has been considered by Prévost [132]. A difficulty with the 3D case compared to the 2D case is that in 3D there are more faces connected to a vertex than there are elements. In 2D, the number of edges and the number of elements connected to a vertex were the same, which allow a straight forward derivation of the linear system. In 3D, additional constraints must be devised to close the system, or the system can be solved in a least-square sense, see [132]. The expansion of fractures for the 3D case is discussed in [75].

4.3 Streamline tracing

Streamlines, pathlines, and streaklines are convenient tools for describing and visualizing flow given by an external velocity field $\boldsymbol{q} = (q_x, q_y, q_z)$. *Streamlines* are a family of curves $\boldsymbol{x} = \boldsymbol{s}(\tau)$ that are instantaneously tangent to the velocity vector $\boldsymbol{q}$ at every point

$$\frac{d\boldsymbol{s}}{d\tau} = \boldsymbol{q}.$$

Streamlines can be traced for any vector field, although the most common is that $\boldsymbol{q}$ represents a velocity obtained from the solution of a set of flow equations. For incompressible flow, streamlines defined at a single instant do not intersect and cannot begin or end inside the fluid. *Streamtubes* are regions bounded by streamlines. Because streamlines are tangent to the velocity field, fluid that is inside a streamtube must remain forever within the same streamtube.

A *pathline* $\boldsymbol{x} = \boldsymbol{p}(t)$ is the trajectory traced out by an imaginary massless particle following the flow of the fluid from a given starting point,

$$\frac{d\boldsymbol{p}}{dt} = \boldsymbol{q}(\boldsymbol{x}, t), \qquad \boldsymbol{p}(t_0) = \boldsymbol{x}_0. \tag{4.33}$$

A *streakline* is the locus at a given instance of the positions of all fluid particles that have gone through a fixed spatial point in the past. In steady flow streamlines, streaklines, and pathlines coincide; in unsteady flow they can be different.

The integration of (4.33) to obtain particle paths and/or travel times, is known as *particle tracking*, for which there exists a rich literature. The particle tracking literature is primarily concerned with problems where the velocity field is only known at a finite set of points, either measured or calculated from a flow model, and interpolation is needed to integrate pathlines. In computational fluid dynamics, particle tracking has been used for visualization [95, 99, 140, 148, 151]. Velocity interpolation in control-volume mixed finite-element methods is a related subject to particle tracking and has been considered in [114]. Within groundwater flow simulation, particle tracking is used to model contaminant transport [28, 52, 125, 139, 143, 145]. In visualization, the integration of (4.33) is usually done numerically using a Runge–Kutta type solver, whereas in groundwater flow, semi-analytical integration is the most common.

In the following we regard *streamline tracing* as a subset of particle tracking, since streamlines may be computed by particle tracking if we introduce the streamline parameter τ as an artificial time variable for which the instantaneous velocity field $\boldsymbol{q}(\boldsymbol{x}, t)$ is steady. In this work we consider streamline tracing in the context of streamline simulation of flow in hydrocarbon reservoirs [22, 33, 94]. In this case, the fluid velocity $\boldsymbol{q}$ is typically given as the numerical solution of a set of flow equations for $\boldsymbol{q}$ and the fluid pressure, e.g., Equation (3.5) or Equation (3.28).

How the corresponding discrete velocity approximation is defined, depends on the numerical method:

- For finite-difference methods, the pressure is usually computed at cell centers, and fluxes can be obtained at cell edges by application of a discrete form of Darcy's law [162].

- For finite-element methods, the numerical solution gives a continuously defined pressure approximation given as the sum of the basis functions for all elements weighted by the corresponding node values. Although a continuously defined velocity can be obtained from Darcy's law, a better strategy is given in [52, 58], where continuous fluxes are obtained at cell edges.

- Mixed finite-element methods *solve* for velocity and pressure simultaneously, resulting in a more accurate velocity field than for finite differences and standard finite elements. The continuously defined velocity is given by the degrees of freedom at the edges and the corresponding basis functions [58, 91]; see also [90, 111, 109].

- Finite-volume methods include multi-point flux approximations [1, 62] and control-volume finite-element methods [35, 154]. In these methods fluxes are computed at cell edges.

In other words, a continuously defined velocity field is obtained only for the mixed finite-element method. For the other methods one must use an interpolation scheme to determine the velocity from the discrete fluxes at the cell edges.

In reservoir simulation and groundwater flow, the predominant way of computing streamlines is by use of a semi-analytical technique. In semi-analytical methods [74, 106, 131, 143], the interpolation of the velocity is simple enough that analytical integration is possible within each grid cell. As an example, let us consider the popular method of Pollock [131]. Given an entry point of a streamline into a grid cell, Pollock's method starts by mapping the grid cell onto the unit square (or unit cube in 3D). Each component of the velocity field is then approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit in each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point and the exit point is mapped back into physical space to give the entry point into the next cell, and so on. In groundwater flow and visualization, more complicated interpolation schemes have been used, where numerical integration is needed, using Euler's method or higher-order Runge-Kutta methods [28, 47, 125, 130, 145].

Tracing of streamlines for use in flow simulations has been investigated in [89, 90, 94, 107, 108, 111, 109, 132, 133]. In this work we focus on irregular grids in three spatial dimensions consisting of hexahedral grid cells with curved surfaces. Streamline tracing may then be performed by a method due to Prévost et al. [133]. This method is a simplification of work done by Cordes and Kinzelbach [52], where Pollock's method is extended to irregular grids. Each grid cell in physical space $\mathcal{P}$ is transformed to a unit cube in a reference space $\mathcal{R}$ using a standard isoparametric trilinear transformation [49]. Next, the velocity in $\mathcal{R}$ is approximated by a linear flux interpolation scaled by an approximation of the Jacobian determinant of the transformation. Finally, the streamline segments passing through each cell can be integrated in $\mathcal{R}$ and mapped back to $\mathcal{P}$. Henceforth, we will call this method *standard flux mapping*, SFM.

As we shall see, the standard flux mapping cannot reproduce a uniform flow field on irregular hexahedrons. Uniform flow or almost uniform flow are important cases to consider, since such flow patterns are likely to occur in large parts of a hydrocarbon reservoir. For example, if the driving force for the flow in the reservoir is gravity, and the permeability of the medium is almost homogeneous, the flow locally is almost uniform in the z-direction of the reservoir. Reproduction

of uniform flow is important if the grid cells are small compared to the variation of the velocity. Also, the failure to reproduce uniform flow will produce errors in the interpolated velocity for the SFM even for nonuniform flow. The error will increase with the irregularity of the grid cells.

To handle the difficulty associated with the standard flux mapping, we consider an alternative method, which we call *corner velocity interpolation* (CVI). Instead of interpolating the velocity field based on discrete fluxes at cell edges, we interpolate directly from point velocities given at the corner points in the grid. This allows for reproduction of uniform flow, and eliminates the influence of the cell geometry on the velocity field. In streamline simulation, we usually only know fluxes, so we also present a method for reconstructing the corner-point velocities in each grid cell from fluxes.

4.3.1 Short literature review

The origin of the literature for streamline tracing for streamline simulation comes from early developments in the field of groundwater simulation. In the paper of Pollock (1988) [131], the flow equations are solved by finite differences on a Cartesian grid. Mass conservative fluxes are obtained at the grid cell boundaries by differencing the discrete pressure solution. Within each grid cell, each component of the velocity was assumed to vary linearly in the one direction, and to be piecewise constant in the other directions. This enabled an analytical integration of the velocity to obtain streamlines and time-of-flight within each cell.

Goode (1990) [74] summarizes the advantages and disadvantages of different interpolation methods on Cartesian grids. The first contribution for unstructured grids and for the finite element method come from Cordes and Kinzelbach (1992) [52]. It is recognized that the finite element method results in discontinuous fluxes, and hence poor-quality streamlines. A flux post-processing is devised that recovers continuous fluxes on a sub grid. The post-processing is relatively cheap and gives improved streamlines in 2D. The extension to hexahedral elements in 3D is discussed, and found to be considerably more complex.

King and Datta-Gupta (1998) [94] discuss the extension of the streamline method to corner-point grids in 3D. Prévost et al. (2002) [133] considered two types of vertex-centered finite volume methods, i.e., where the permeability is constant over each element (the control-volume finite element method), and where the permeability is constant over each polygon in the dual mesh (flux continuous scheme) for the solution of the flow equations. For such a method, fluxes are computed at the edges polygons. However, streamline generation requires mesh elements that are triangles or quadrilaterals (tetrahedrons or hexahedrons in 3D). The post-processing of Cordes and Kinzelbach was now further developed as a method to generate fluxes on a sub grid consisting of triangles or quadrilaterals.

For the tracing in 3D it was proposed that the Jacobian of the trilinear transformation of a hexahedron to a unit cube was evaluated at the midpoint of the cell in order to avoid a numerical integration of the velocity field.

Prévost (2003) [132] further develops the flux recovery in [133]. The recovery for the 3D case is shown to be considerably more complex than for the 2D case. Matringe et al. (2004) [108] considers the effect of the launching location on the accuracy of streamlines. An adaptive mesh refinement procedure on Cartesian grids is proposed for improving the accuracy of the velocity.

Since the topic of flux recovery is closely related to streamline tracing, we mention three recent papers on this topic. Sun and Wheeler (2006) [152] considered flux recovery for non-conservative methods, like the continuous Garlerkin method and Correa and Loula (2007) [53] and Cockburn et al. (2007) [50] also considered flux recovery for the continuous Galerkin method.

Matringe et al. (2006) [109], Matringe et al. (2007) [111] and Matringe et al. (2006) [110] considers the mixed finite element method on triangular and quadrilateral grids. The divergence free BDM_1 elements are investigated. The method is applicable when solving for two unknowns per edge in addition to the pressure in the mixed finite element method, or when using the MPFA method. In the latter case, only the cell-center pressures are solved for, but still two fluxes can be computed for each edge in terms of the pressures. The streamlines are given analytically by a streamfunction within each cell. However, the exit point and time-of-flight must be computed numerically.

Jimenez et al. (2007) [89] considered how to improve the time-of-flight calculations in streamline tracing. By reformulating the streamline differential equation, they were able to integrate the equations analytically without approximating the Jacobian for irregular grid cells. This enabled exact reproduction of uniform flow on irregular grids in 2D.

Hægland et al. (2007) [77] considered the problem of reproducing uniform flow on three dimensional irregular hexahedral grids. The new interpolation scheme could reproduce uniform flow, but the velocity had to be integrated numerically and point-wise continuity of the normal component of the velocity at the grid cell faces was lost.

Juanes and Matringe (2008) [90] considers the mixed finite element method on unstructured grids in 2D. They identify high-order (BDM) velocity spaces that induce a stream function, and show the potential improvements to be expected using such a high-order method.

4.3.2 Introduction: Tracing on Cartesian Grids

To motivate the description of streamline tracing on irregular grids, we start now by discussing the basic version on Cartesian grids, which is commonly referred

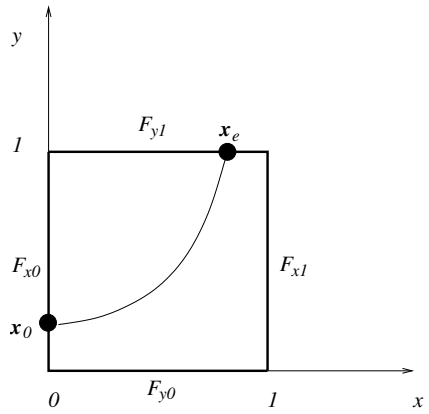


Figure 4.8: Streamline tracing on a unit square.

to as Pollock's method in the literature. As we saw in the introduction, Pollock's method builds a streamline as a series of (small) line segments that each cross a grid cell in physical space. The segments are constructed such that the exit point of the streamline in one cell is the entrance point in the next cell. For the development in herein, it is sufficient to consider the method in the unit square (or unit cube in 3D).

Pollock's Method

The method will be presented for 2D; the extension to 3D is straightforward. Linear interpolation of the edge fluxes is then used to define a velocity field (see Figure 4.8)

$$\boldsymbol{q}^{\mathrm{I}}(x,y) \equiv \begin{bmatrix} F_{x0}(1-x)+F_{x1}x \\ F_{y0}(1-y)+F_{y1}y \end{bmatrix}, \qquad 0 \le x \le 1, \quad 0 \le y \le 1. \tag{4.34}$$

Here the superscript I refers to the fact that the velocity field is interpolated based on fluxes given at the edges, as shown in Figure 4.8.

Having defined a velocity field, the streamline $\boldsymbol{s}(t) = \boldsymbol{x}(t) = [x(t), y(t)]$ is found by integrating the system of ODEs in (4.33):

$$\begin{cases} \dfrac{dx}{dt} = q_x^{\mathrm{I}}(x), & x(0) = x_0, \\ \dfrac{dy}{dt} = q_y^{\mathrm{I}}(y), & y(0) = y_0, \end{cases} \tag{4.35}$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$. Since q_x^I depends only on x, and q_y^{I} depends only on y, the streamline can be found analytically [131]: Assuming $F_{x0} \neq F_{x1}$ and $F_{y0} \neq F_{y1}$, integration of each of the equations in (4.35) yields

two separate expressions for the travel time along the streamline as a function of x and y, respectively,

$$t_x = \frac{1}{F_{x1} - F_{x0}} \ln\left(\frac{q_x^{\mathrm{I}}(x)}{q_x^{\mathrm{I}}(x_0)}\right), \tag{4.36}$$

$$t_y = \frac{1}{F_{y1} - F_{y0}} \ln\left(\frac{q_y^{\mathrm{I}}(y)}{q_y^{\mathrm{I}}(y_0)}\right). \tag{4.37}$$

By inserting $x = 0$ and $x = 1$ in (4.36) and $y = 0$ and $y = 1$ in (4.37), we determine the times t_{x0}, t_{x1}, t_{y0}, and t_{y1}, respectively, when the streamline crosses the corresponding straight lines. (Notice that these times may be negative or infinite). From these four travel times we can easily determine the exit time t_e when the streamline leaves the unit square. The exit point $\boldsymbol{x}_e$ is then found by inserting t_e in (4.36) and (4.37).

Applying Pollock's Method in Reservoir Simulation

For flow in porous media, the flow velocity $\boldsymbol{q}$ is obtained by solving, e.g., Equation (3.28) using, e.g., a flux continuous scheme [1] to provide fluxes on each grid cell edge. In order to obtain particle velocities, these fluxes should be divided by the porosity. We assume for the moment that the grid cells can be any quadrilateral. The fluxes will be used to define a velocity field $\boldsymbol{q}^{\mathrm{I}}$ that approximates $\boldsymbol{q}$.

The flux is the integral of the normal component of the velocity field across an edge. We will require that $\boldsymbol{q}^{\mathrm{I}}$ reproduces the given fluxes,

$$F_E = \int_E \boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{\nu}_E ds, \tag{4.38}$$

where the subscript E refers to the edge, F_E is the flux over the edge, and $\boldsymbol{\nu}_E$ is a unit normal to the edge. At the common edge between two adjacent grid cells, the absolute value of the flux is the same seen from both cells. By defining $\boldsymbol{\nu}_E$ suitably, we can assure that the sign of the flux is also the same. Thus only one flux is needed per cell edge.

We next assume that the normal component $(\boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{\nu}_E)$ is constant along a given edge. Then (4.38) becomes

$$F_{E_i} = (\boldsymbol{q}^{\mathrm{I}}|_{E_i} \cdot \boldsymbol{\nu}_{E_i})\,|E_i|, \qquad i = 1,2,3,4, \tag{4.39}$$

where E_i is one of the four edges, $\boldsymbol{q}^{\mathrm{I}}|_{E_i}$ denotes $\boldsymbol{q}^{\mathrm{I}}$ evaluated at a point on the edge, and $|E_i|$ is the length of the edge. Using this, combined with (4.38) for a unit square, gives

$$q_x^{\mathrm{I}}(0,y) = F_{x0}, \qquad q_y^{\mathrm{I}}(x,0) = F_{y0},$$
$$q_x^{\mathrm{I}}(1,y) = F_{x1}, \qquad q_y^{\mathrm{I}}(x,1) = F_{y1},$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$, respectively. We are now in a position to introduce the velocity interpolation (4.34) and use Pollock's method to obtain the streamlines, as described above.

Finally, note that the interpolation step is not always necessary since (4.34) can be obtained directly when solving the pressure equation with a mixed finite-element method using the lowest-order Raviart-Thomas (RT_0) elements [34].

4.3.3 Streamline and Velocity in Curvilinear Coordinates

Streamline tracing on irregular grid cells involves introducing a curvilinear coordinate system [100] for each grid cell. In this section we describe the coordinate transformation and how to express the velocity in curvilinear coordinates. This transformation of coordinates and velocity will be fundamental for the development of the CVI method. Additionally, it will be used to derive the SFM and EFM methods.

Consider a quadrilateral grid cell in physical space $\mathcal{P}$ given by the four corner points $\boldsymbol{x}_i = [x_i, y_i]$, $i = 1, \ldots 4$. By using the bilinear isoparametric transformation [49, 52, 133]

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) \equiv \sum_{i=1}^{4} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}), \tag{4.40}$$

each grid cell is transformed into a unit square in the reference space $\mathcal{R}$. Here $\hat{\boldsymbol{x}} = [\hat{x}, \hat{y}]$ is a point in $\mathcal{R}$; $\boldsymbol{x}(\hat{\boldsymbol{x}}) = [x(\hat{x}, \hat{y}), y(\hat{x}, \hat{y})]$ is a point in physical space $\mathcal{P}$; and $\phi_i(\hat{x}, \hat{y})$, $i = 1, \ldots 4$, are the standard bilinear shape functions on the unit square.

Later we will compute normal vectors to the cell edges. In order to obtain a well-defined direction of these normals we require that $\boldsymbol{x}_i$, $i = 1, \ldots, 4$, are the logically bottom-left, bottom-right, top-left, and top-right corner of the quadrilateral, respectively.

A Velocity Transformation

If we can describe the velocity in bilinear coordinates, the streamline can be integrated in bilinear coordinates, and since each grid cell is a unit square in $\mathcal{R}$, Pollock's method is applicable. To obtain the streamline $\boldsymbol{s}(t)$ in $\mathcal{P}$, the bilinear transformation is applied to the streamline $\hat{\boldsymbol{s}}(t)$ in $\mathcal{R}$; see Figure 4.9. From this we can use the chain rule to deduce the velocity in $\mathcal{R}$,

$$\boldsymbol{q} \equiv \frac{d\boldsymbol{s}}{dt} = \frac{d\boldsymbol{x}(\hat{\boldsymbol{s}}(t))}{dt} = \frac{d\boldsymbol{x}}{d\hat{\boldsymbol{x}}} \frac{d\hat{\boldsymbol{s}}}{dt} = \boldsymbol{J}\hat{\boldsymbol{q}}. \tag{4.41}$$

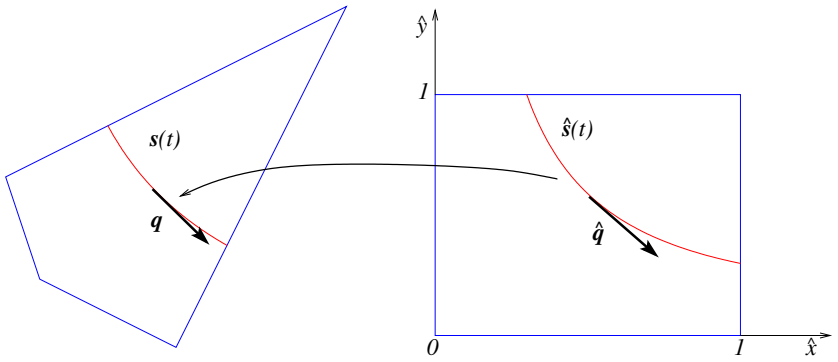


Figure 4.9: Transformation of a streamline and velocity from reference space $\mathcal{R}$ to physical space $\mathcal{P}$.

Here $\hat{\boldsymbol{q}} = d\hat{\boldsymbol{s}}/dt$ is the velocity in $\mathcal{R}$, $\boldsymbol{q}$ is the velocity in $\mathcal{P}$, and $\boldsymbol{J} = d\boldsymbol{x}/d\hat{\boldsymbol{x}}$ is the Jacobian matrix of the transformation. Thus, the transformed velocity is given by

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q}. \tag{4.42}$$

Note that the Piola transformation [34, 90] of a vector field given by

$$\hat{\boldsymbol{q}}^{\mathrm{P}} = (\det \boldsymbol{J})\boldsymbol{J}^{-1}\boldsymbol{q} \tag{4.43}$$

is constructed so that fluxes are preserved in reference space. By comparing (4.42) and (4.43) we see that $\hat{\boldsymbol{q}}^{\mathrm{P}} = (\det \boldsymbol{J})\hat{\boldsymbol{q}}$.

4.3.4 Extending Pollock's Method to Irregular Grids

Tracing on irregular grids is done in reference space $\mathcal{R}$ using (4.42). The Jacobian matrix is given by

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \hat{x}} & \dfrac{\partial x}{\partial \hat{y}} \\ \dfrac{\partial y}{\partial \hat{x}} & \dfrac{\partial y}{\partial \hat{y}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_x & \boldsymbol{u}_y \end{bmatrix}, \tag{4.44}$$

where $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are the base vectors of the bilinear coordinates. Thus, the inverse of the Jacobian matrix can be expressed in terms of contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, as

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \end{bmatrix}, \tag{4.45}$$

where

$$\boldsymbol{n}_x = [\partial y/\partial \hat{y}, -\partial x/\partial \hat{y}]^{\mathrm{T}}, \qquad \text{and} \qquad \boldsymbol{n}_y = [-\partial y/\partial \hat{x}, \partial x/\partial \hat{x}]^{\mathrm{T}}. \tag{4.46}$$

The contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, are normal vectors to edges of the quadrilateral in physical space $\mathcal{P}$. These vectors are generally not constant, but when evaluated at a particular edge of the quadrilateral, they are constant; and the length of these vectors is then equal to the length of the edge. It follows from (4.42) and (4.45) that,

$$\hat{\boldsymbol{q}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q} \cdot \boldsymbol{n}_x \\ \boldsymbol{q} \cdot \boldsymbol{n}_y \end{bmatrix}. \tag{4.47}$$

Next, we approximate $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ by $\boldsymbol{q}^{\mathrm{I}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}$, respectively, based on the given fluxes in $\mathcal{P}$. Still using (4.39) for $\boldsymbol{q}^{\mathrm{I}}$, and recognizing the dot products in (4.47) as fluxes since $\boldsymbol{n}_E = \boldsymbol{\nu}_E |E|$, we can define $\hat{\boldsymbol{q}}^{\mathrm{I}}$ by the use of a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}. \tag{4.48}$$

This expression is the basis for the standard flux-mapping (SFM) method [133] and the extended flux-mapping (EFM) method of Jimenez et al. [89]. In the SFM method [133], the bilinear Jacobian determinant is replaced by a constant value in order to make analytical integration possible. Hence,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{SFM}} = \frac{1}{\det \boldsymbol{J}_m} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}, \tag{4.49}$$

where $\boldsymbol{J}_m = \boldsymbol{J}(0.5, 0.5)$ is the Jacobian matrix evaluated at the midpoint of the reference element. In [114] it was shown that (4.48) is exact for uniform flow in 2D. Since the Jacobian determinant only scales the absolute value of the velocity in (4.49), SFM reproduces the shape of the streamlines exactly for uniform flow. However, approximating the Jacobian determinant by a constant introduces errors in computing time-of-flight, as noted by Jimenez et al. [89]. They demonstrated that by using a pseudo time-of-flight τ, the velocity in (4.48) can be integrated analytically by rewriting (4.48) as

$$\begin{cases} \dfrac{d\hat{x}}{F_{x0}(1-\hat{x}) + F_{x1}\hat{x}} = \dfrac{dt}{\det \boldsymbol{J}} = d\tau, \\ \dfrac{d\hat{y}}{F_{y0}(1-\hat{y}) + F_{y1}\hat{y}} = \dfrac{dt}{\det \boldsymbol{J}} = d\tau. \end{cases} \tag{4.50}$$

Then Pollock's method is used to find $\hat{x}(\tau)$ and $\hat{y}(\tau)$ and the exit pseudo time τ_e. To find the real exit time, t_e, we integrate the determinant of the Jacobian

$$t_e = \int_0^{t(\tau_e)} dt = \int_0^{\tau_e} \det \boldsymbol{J}(\hat{x}(\tau), \hat{y}(\tau)) d\tau. \tag{4.51}$$

Thus, the EFM method is characterized by the velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}}$ given in (4.48), where the Jacobian determinant is evaluated exactly.

4.3.5 Extending the SFM and EFM Methods to 3D

We consider an irregular grid consisting of hexahedral grid cells with fluxes computed on the faces of each cell. Each hexahedron in physical space $\mathcal{P}$ will be defined as a one-to-one trilinear map [49, 52, 133] of a unit cube in a reference space $\mathcal{R}$. The trilinear map is given by

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) = \sum_{i=1}^{8} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}, \hat{z}), \tag{4.52}$$

where $\boldsymbol{x}_i$, $i = 1, \ldots, 8$ are the coordinates of the eight corners defining the grid cell, and $\phi_i(\hat{x}, \hat{y}, \hat{z})$, $i = 1, \ldots, 8$ are the standard trilinear shape functions on the unit cube. Note that these hexahedrons generally have curved surfaces.

We remark that the mapping gives a natural definition of the faces of the hexahedron in physical space. We define general x-, y-, and z-surfaces in $\mathcal{P}$ by

$$\begin{aligned} s_x(\hat{x}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{y} \le 1, 0 \le \hat{z} \le 1\}, \\ s_y(\hat{y}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{x} \le 1, 0 \le \hat{z} \le 1\}, \\ s_z(\hat{z}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \le \hat{x} \le 1, 0 \le \hat{y} \le 1\}. \end{aligned} \tag{4.53}$$

For the primary faces, $s_x(0)$, $s_x(1)$, etc., we will also use the notation S_{x0}, S_{x1}, respectively.

The velocity in physical space $\mathcal{P}$ is related to the velocity in reference space $\mathcal{R}$ by (4.41). The Jacobian matrix of the transformation is written,

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \hat{x}} & \dfrac{\partial x}{\partial \hat{y}} & \dfrac{\partial x}{\partial \hat{z}} \\ \dfrac{\partial y}{\partial \hat{x}} & \dfrac{\partial y}{\partial \hat{y}} & \dfrac{\partial y}{\partial \hat{z}} \\ \dfrac{\partial z}{\partial \hat{x}} & \dfrac{\partial z}{\partial \hat{y}} & \dfrac{\partial z}{\partial \hat{z}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_x & \boldsymbol{u}_y & \boldsymbol{u}_z \end{bmatrix}. \tag{4.54}$$

Here $\boldsymbol{u}_x$, $\boldsymbol{u}_y$, and $\boldsymbol{u}_z$ are the covariant base vectors of the trilinear coordinates. It follows that the inverse of $\boldsymbol{J}$ can be expressed in terms of contravariant vectors $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, and $\boldsymbol{n}_z$ such that

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \\ \boldsymbol{n}_z^{\mathrm{T}} \end{bmatrix}, \tag{4.55}$$

where

$$\boldsymbol{n}_x = \boldsymbol{u}_y \times \boldsymbol{u}_z, \qquad \boldsymbol{n}_y = \boldsymbol{u}_z \times \boldsymbol{u}_x, \qquad \boldsymbol{n}_z = \boldsymbol{u}_x \times \boldsymbol{u}_y. \tag{4.56}$$

Therefore,

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q}\cdot\boldsymbol{n}_x \\ \boldsymbol{q}\cdot\boldsymbol{n}_y \\ \boldsymbol{q}\cdot\boldsymbol{n}_z \end{bmatrix}. \tag{4.57}$$

As in the 2D case, the EFM method of Jimenez et al. [89] is based on (4.57) in combination with a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \\ F_{z0}(1-\hat{z}) + F_{z1}\hat{z} \end{bmatrix}, \tag{4.58}$$

where F_{x0} is the given flux over the face S_{x0} in $\mathcal{P}$ corresponding to $\hat{x}$=0, etc. We will later use the normal vectors at the six primary faces defined as,

$$\boldsymbol{n}_{x0}(\hat{y},\hat{z}) \equiv \boldsymbol{n}_x(0,\hat{y},\hat{z}), \qquad \boldsymbol{n}_{x1}(\hat{y},\hat{z}) \equiv \boldsymbol{n}_x(1,\hat{y},\hat{z}), \tag{4.59}$$

$$\boldsymbol{n}_{y0}(\hat{x},\hat{z}) \equiv \boldsymbol{n}_y(\hat{x},0,\hat{z}), \qquad \boldsymbol{n}_{y1}(\hat{x},\hat{z}) \equiv \boldsymbol{n}_y(\hat{x},1,\hat{z}), \tag{4.60}$$

$$\boldsymbol{n}_{z0}(\hat{x},\hat{y}) \equiv \boldsymbol{n}_z(\hat{x},\hat{y},0), \qquad \boldsymbol{n}_{z1}(\hat{x},\hat{y}) \equiv \boldsymbol{n}_z(\hat{x},\hat{y},1), \tag{4.61}$$

where $(\hat{x},\hat{y},\hat{z}) \in [0,1]\times[0,1]\times[0,1]$.

In the SFM method by Prévost et al. [133], the Jacobian in (4.58) is evaluated at the midpoint of the unit cube.

4.3.6 Reproduction of Uniform Flow

By uniform flow we refer to flow given by a constant velocity field $\boldsymbol{q}$. Obviously, uniform flow leads to straight and parallel streamlines, and therefore the time-of-flight is equal at all points having the same distance from the inflow boundary.

In 2D, the linear flux interpolation used by EFM is exact for uniform flow, as shown in [114]. However, this is not the case in 3D: in [114] it was shown that the flux of a uniform flow field will vary quadratically. This might lead one to believe that replacing the linear interpolation in (4.58) with a quadratic, would solve the problem.

The difficulty with any flux interpolation can be seen from the following argument: Consider for instance the surface S_{x0}. A normal vector $\boldsymbol{n}_{x0}$ to this face at $\boldsymbol{x}(0,\hat{y},\hat{z})$ is given in (4.59). The absolute value of $\boldsymbol{n}_{x0}$ equals the surface Jacobian, which in the case of planar faces is constant only for parallelograms. For non-planar faces the direction of the normal vector is not constant either. By inserting $\hat{x}$=0 in (4.57) and (4.58) we see that both SFM and EFM use the following approximation

$$\boldsymbol{q}\cdot\boldsymbol{n}_{x0} = F_{x0}. \tag{4.62}$$

For uniform flow, $\boldsymbol{q}$ is constant but $\boldsymbol{n}_{x0}$ will generally not be a constant, as noted above. The normal vector in (4.62) cannot change at a fixed point, so in effect we will trace the streamline using a velocity $\tilde{\boldsymbol{q}}$ such that $\tilde{\boldsymbol{q}} \cdot \boldsymbol{n}_{x0} = F_{x0}$, and $\tilde{\boldsymbol{q}}$ compensates for the fact that $\boldsymbol{n}_{x0}$ is not constant. Hence, the tracing velocity $\tilde{\boldsymbol{q}}$ will depend on the normal vector $\boldsymbol{n}_{x0}$, or in other words, depend on the geometry of the cell. Therefore uniform flow cannot be reproduced.

4.4 Corner Velocity Interpolation

In the previous section we described how the SFM and the EFM methods fail to reproduce uniform flow on e.g., grids with nonplanar faces. To remedy this problem, we will propose a different velocity interpolation scheme, which we will denote *corner velocity interpolation* (CVI). For simplicity, the method will first be introduced in 2D and then extended to 3D in Section 4.4.2.

4.4.1 Interpolation in 2D

As for the SFM and EFM methods introduced in Section 4.3.4, we will formulate the CVI method using a cell-by-cell integration in the unit cube in reference space $\mathcal{R}$. However, the method may also be used to integrate streamlines directly in physical space; see [76] for more details. To formulate the CVI method, we start with the relation (4.47) for the velocity in $\mathcal{R}$, where the unknown quantity is the velocity $\boldsymbol{q}$ in $\mathcal{P}$. We approximate $\boldsymbol{q}$ by a bilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1, \ldots, 4$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(\boldsymbol{x}(\hat{x}, \hat{y})) \equiv \sum_{i=1}^{4} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}). \tag{4.63}$$

The corner velocities $\boldsymbol{q}_i$ will be reconstructed from the given fluxes, such that $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ is exact for uniform flow. This means that all $\boldsymbol{q}_i$ will be equal if $\boldsymbol{q}$ is constant.

Hence the CVI method is characterized by the following velocity interpolation in $\mathcal{R}$,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \boldsymbol{J}^{-1} \boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}. \tag{4.64}$$

Reconstruction of Corner Velocities

We consider the cell shown in Figure 4.10. The four fluxes F_i, will be given on the edges E_i for $i = x0, x1, y0, y1$, and the normal vectors are defined in (4.46). Note that $\boldsymbol{n}_x(\hat{x}, \hat{y}) = \boldsymbol{n}_x(\hat{x})$ and $\boldsymbol{n}_y(\hat{x}, \hat{y}) = \boldsymbol{n}_y(\hat{y})$. Hence, we can define $\boldsymbol{n}_{x0} = \boldsymbol{n}_x(0)$, $\boldsymbol{n}_{x1} = \boldsymbol{n}_x(1)$, $\boldsymbol{n}_{y0} = \boldsymbol{n}_y(0)$, and $\boldsymbol{n}_{y1} = \boldsymbol{n}_y(1)$, respectively.

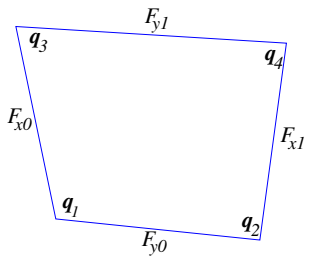


Figure 4.10: Reconstructing velocities from fluxes in 2D

The corner velocities $\boldsymbol{q}_i$, $i = 1, \ldots, 4$, will be solutions of 2×2 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ex(i)} = F_{Ex(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ey(i)} = F_{Ey(i)}, \end{cases} \qquad i = 1, \ldots, 4. \tag{4.65}$$

Here $Ex(i)$ and $Ey(i)$, $i = 1, \ldots, 4$, refer to edges in the x- and y-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. This means that for $\boldsymbol{q}_1$, we get

$$\begin{cases} \boldsymbol{q}_1 \cdot \boldsymbol{n}_{x0} = F_{x0}, \\ \boldsymbol{q}_1 \cdot \boldsymbol{n}_{y0} = F_{y0}, \end{cases} \tag{4.66}$$

since E_{x0} and E_{y0} are adjacent to corner $\boldsymbol{x}_1$. The systems (4.65) are well-conditioned as long as the quadrilateral does not degenerate. If the fluxes have been computed exactly for a uniform flow field $\boldsymbol{q}$, then $\boldsymbol{q}_i = \boldsymbol{q}$.

Note that (4.65) implies that

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i) \cdot \boldsymbol{n}_i = F_i, \qquad i = x0, x1, y0, y1, \tag{4.67}$$

where $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i)$ denotes $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ evaluated at a point on edge E_i. Hence, since the length of $\boldsymbol{n}_i$ is equal to the length of E_i, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ will reproduce the given edge fluxes.

4.4.2 Extension to 3D

We approximate $\boldsymbol{q}$ in (4.57) by a trilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1, \ldots, 8$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \sum_{i=1}^{8} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}, \hat{z}). \tag{4.68}$$

Reconstructing Corner Velocities

Consider the cell in Figure 4.11. The six fluxes F_i will be given on the faces S_i for $i = x0, x1, y0, y1, z0, z1$, respectively. The corresponding normal vectors $\boldsymbol{n}_i$ are defined in (4.59).

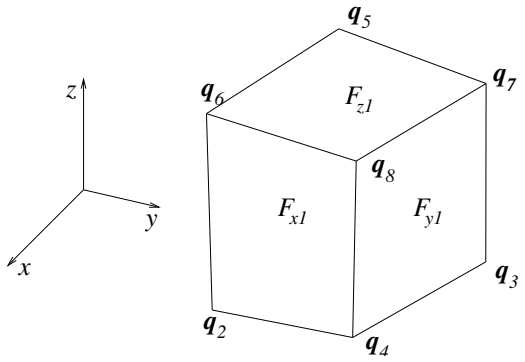


Figure 4.11: Reconstructing velocities from fluxes in 3D

The flux integral of a velocity field $\boldsymbol{q}$ over the face S_i can be transformed to a double integral on a face of the unit cube in $\mathcal{R}$ using the trilinear transformation:

$$F_i = \int_0^1 \int_0^1 \boldsymbol{q} \cdot \boldsymbol{n}_i d\alpha d\beta, \qquad i = x0, x1, y0, y1, z0, z1, \tag{4.69}$$

since the norm of $\boldsymbol{n}_i$ is equal to the surface Jacobian. If $\boldsymbol{q}$ is constant, we get,

$$F_i = \boldsymbol{q} \cdot \int_0^1 \int_0^1 \boldsymbol{n}_i(\alpha, \beta) d\alpha d\beta = \boldsymbol{q} \cdot \bar{\boldsymbol{n}}_i, \qquad i = x0, x1, y0, y1, z0, z1, \tag{4.70}$$

where $\bar{\boldsymbol{n}}_i$, defined by the above equation, is given by a simple analytic expression, see e.g., [1]. Thus, in order to reproduce a uniform flow field, the corner velocities $\boldsymbol{q}_i$, $i = 1, \ldots, 8$, will be solutions of 3×3 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sx(i)} = F_{Sx(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sy(i)} = F_{Sy(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sz(i)} = F_{Sz(i)}, \end{cases} \qquad i = 1, \ldots, 8. \tag{4.71}$$

Here $Sx(i)$, $Sy(i)$, and $Sz(i)$, $i = 1, \ldots, 8$, refer to faces in the x-, y-, and z-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. That is, for $\boldsymbol{q}_8$ we get

$$\begin{cases} \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{x1} = F_{x1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{y1} = F_{y1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{z1} = F_{z1}. \end{cases} \tag{4.72}$$

Now, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ are given by (4.68) and (4.64), respectively. As opposed to SFM and EFM, each component of the interpolated velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is a function of all three variables $\hat{x}$, $\hat{y}$, and $\hat{z}$. Therefore, analytical integration of $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is generally not possible.

Note that numerical integration of a velocity in $\mathcal{R}$ will not give the exact exit point from the cell, unless the last integration step ends on the cell boundary. Generally, interpolation is needed to determine the exit point [130].

The CVI-interpolation and reproduction of uniform flow

The main feature of the CVI-interpolation scheme is that it can reproduce uniform flow. This means that the CVI-interpolation will resolve the constant velocity fields on irregular hexahedral grids in 3D. We note that this is not the case for standard interpolation spaces, e.g., used in the mixed finite element setting.

We show that the CVI velocity field $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ given Equation (4.68) reproduces a uniform flow field $\boldsymbol{v}$, when the fluxes agree with $\boldsymbol{v}$. Hence, the fluxes are assumed to be given as

$$F_i = \int_0^1 \int_0^1 \boldsymbol{v} \cdot \boldsymbol{n}_i(\alpha, \beta) d\alpha d\beta = \boldsymbol{v} \cdot \int_0^1 \int_0^1 \boldsymbol{n}_i(\alpha, \beta) d\alpha d\beta = \boldsymbol{v} \cdot \bar{\boldsymbol{n}}_i, \qquad i = x0, x1, y0, y1, z0, z1. \tag{4.73}$$

It is sufficient to show that $\boldsymbol{q}_i \equiv \boldsymbol{v}$, $i = 1, \ldots, 8$, for then, according to Equation (4.68)

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \sum_{i=1}^{8} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}, \hat{z}) = \sum_{i=1}^{8} \boldsymbol{v} \phi_i = \boldsymbol{v} \sum_{i=1}^{8} \phi_i = \boldsymbol{v}, \tag{4.74}$$

since $\sum_{i=1}^{8} \phi_i = 1$. We will show that $\boldsymbol{q}_1 = \boldsymbol{v}$, the derivation is similar for the other corner velocities. By definition (see Equation (4.71)), $\boldsymbol{q}_1$ satisfies the following equations

$$\begin{cases} \boldsymbol{q}_1 \cdot \bar{\boldsymbol{n}}_{x0} = F_{x0}, \\ \boldsymbol{q}_1 \cdot \bar{\boldsymbol{n}}_{y0} = F_{y0}, \\ \boldsymbol{q}_1 \cdot \bar{\boldsymbol{n}}_{z0} = F_{z0}. \end{cases} \tag{4.75}$$

From Equation (4.73), $\boldsymbol{v}$ satisfies exactly the same conditions,

$$\begin{cases} \boldsymbol{v} \cdot \bar{\boldsymbol{n}}_{x0} = F_{x0}, \\ \boldsymbol{v} \cdot \bar{\boldsymbol{n}}_{y0} = F_{y0}, \\ \boldsymbol{v} \cdot \bar{\boldsymbol{n}}_{z0} = F_{z0}. \end{cases} \tag{4.76}$$

The result follows since the normal vectors $\bar{\boldsymbol{n}}_{x0}$, $\bar{\boldsymbol{n}}_{y0}$, and $\bar{\boldsymbol{n}}_{z0}$ are linearly independent and constitute a basis for $\mathbb{R}^3$.

Chapter 5

Ongoing and further work

In this chapter we will discuss some ongoing work. The first section discusses a possible extension of the CVI-method (Chapter 4.3 and [77]), to avoid the problem with discontinuity in the normal component at cell faces, as shown in [124]. The last section consider application of streamline tracing to two-phase flow with a diffusion term in the transport equation. This problem is relevant to, for example, vertically averaged flow of CO_2 and water in a CO_2 sequestration scenario. Cross-streamline effects in the transport equation, like capillary pressure, are usually handled by operator splitting, see [55] and references therein, involving a mapping from streamlines to a background grid within each operator splitting time step. We propose here a method that avoids the problems associated with the mapping and also speeds up the solution, by introducing normal lines, i.e., lines that are perpendicular to the streamlines.

5.1 Improved CVI method

We consider the point-wise reconstruction of velocity fields on general hexahedral grids based on flux interpolation. In [77] we developed a streamline tracing method that could reproduce uniform flow on a general hexahedral grid, see also Chapter 4.3. The method also had point-wise continuity of the normal component of the velocity across planar cell faces. However, this property is not, in general, fulfilled across curved cell faces, as shown in [124].

Basically, it is shown in [124] that

- If cell interfaces are nonplanar the CVI method loses point-wise continuity;
- When we do not have point-wise continuity, streamlines may terminate abruptly;

- In general it is not possible to construct a piecewise continuous flow field based on only six cell fluxes and at the same time reproduce uniform flow and have a H(div)-conforming velocity field.

A possible way to circumvent this restriction, is to introduce more degrees of freedom into the velocity field, i.e., we consider not only the six local cell face fluxes, but also fluxes of neighboring cells. We briefly outline a method to obtain a reconstructed flow field with the following properties:

- reproduction of uniform flow;
- reproduction of fluxes;
- point-wise continuity of normal component. This guarantees that streamlines do not terminate abruptly;
- zero divergence for cells with no sinks or sources,
- rotation invariant velocity space.

In order to achieve this, we introduce a local space for the three normal components of the velocity field. The normal components are taken with respect to the contravariant vectors of the hexahedron, see, e.g., the vector on the right hand side of Equation (4.57). Each of the three normal components are represented as a polynomial with 9 degrees of freedom. Hence, there will be a total of 27 degrees of freedom (DOFs) per hexahedron. It can be shown that 3 DOFs are associated with each face of the hexahedron, making a total of 18 DOFs associated with faces, and the remaining 9 DOFs are associated with the interior.

The six available DOFs (the six cell face fluxes) from the pressure solution are incorporated by requiring that the reconstructed flow field, as given similarly to Equation (4.57), should reproduce these fluxes. Hence, there remain two DOFs per face to be determined. The important observation now is that these two conditions, on each face, have to be determined exactly the same way for each of the two hexahedrons sharing that face, in order to obtain point-wise continuity of the normal component of the velocity. At the same time, these DOFs should be consistent with a uniform flow field. This lead us to consider the original CVI-velocity field, as discussed in Chapter 4.3, since this velocity field is known to reproduce a uniform flow field. Since the CVI-velocity field is locally reconstructed, there will in general be two different CVI-normal components at a point on the face, one for each of the two cells sharing that face. By taking the harmonic average of these two components, we define a new quantity that will be the same seen from each of the two cells. We will require that the normal component of the improved velocity (we call this velocity for CVIE to distinguish it from the CVI-velocity)

is equal to this harmonic average at a set of points on the interface. As there are only two DOFs left to be determined for the CVIE normal component at the face, two points will be sufficient. However, it is not possible to choose two point symmetrically at the face, so we propose to choose, e.g., four symmetrical points, and minimize the deviation in a least-square sense, at these points.

The last step is to determine the 9 internal DOFs. By requiring that the CVIE-velocity should have a constant divergence equal to the sum of the six cell face fluxes, 7 DOFs are resolved. Note that this means that the velocity is divergence free for cells with no sources or sinks. The seven conditions on the internal DOFs are set up such that the divergence associated with the 18 exterior DOFs cancels the divergence associated with the interior DOFs. The last two internal DOFs can be determined by minimizing the deviation from a CVI-like velocity field at some symmetric points in the interior of the hexahedron. We omit the details.

5.2 Vertical averaging and normal lines

In general, flow through a porous medium is three-dimensional. However, since for many aquifers the geometry is such that they are thin relative to their horizontal dimensions, a simpler approach may be used where we assume that the flow in the aquifer is everywhere essential horizontal, neglecting vertical flow components. Then two-dimensional flow and transport equations are derived by integrating the three-dimensional equations along the height of the aquifer. This approach have been used extensively in the groundwater literature, and is called the hydraulic approach [24].

In Nordbotten and Celia [122] an analytical solution for injection of CO_2 into deep, confined saline aquifers was derived. The analytical solution captured the system behavior well, under the assumptions of radial horizontal flow in a homogeneous aquifer. The purpose of this section is to remove the assumptions of radial symmetry and a homogeneous aquifer. However, since in this case the resulting equations cannot be solved analytically, an efficient numerical solution procedure is important.

We consider incompressible flow of two immiscible fluids in a three-dimensional domain Ω of constant thickness H. Further, we assume that one of the fluids is always on top of the other fluid and that there is a sharp interface between the fluids (no mixing), see [122] for further details.

At any time t the interface between the fluids, $h(x, y, t)$, divides Ω into two connected domains. By assuming a hydrostatic pressure distribution in the vertical direction and integrating the mass conservation equation for each fluid in the same direction, we will arrive at a new set of two-dimensional equations with the pressure of one of the phases and the height of the interface h as unknowns.

The set of equations consists of one pressure equation of elliptic character and one parabolic equation for the evolution of the interface height, or the averaged saturation of the bottom fluid S across the vertical. Assuming an isotropic permeability tensor, the saturation equation can be written

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (f(S)\boldsymbol{v} + g(S, \boldsymbol{x})\nabla S) = q. \tag{5.1}$$

In the above equation, ϕ is the porosity, $f(S)$ is a fractional flow type function incorporating effects of viscosity differences between the two fluids, $\boldsymbol{v}$ is the sum of the averaged Darcy velocities of each fluid, $g(S, \boldsymbol{x})$ is a function incorporating effects like permeability, density differences, and viscosity differences, and q is the sum of the averaged source terms from both fluids.

We will assume a sequential solution procedure [55] where the saturation is assumed fixed when solving the pressure equation, and the total velocity $\boldsymbol{v}$ is assumed fixed when solving the Saturation equation (5.1). For the solution of Equation (5.1) we will use operator splitting combined with streamlines and normal lines, that will reduce the dimension of the equations from 2D to 1D. The splitting will be done such that advective effects will be solved along the streamlines, and diffusive effects will be solved along both streamlines and normal lines. Alternative splittings, like dimensional splitting, have been discussed in, e.g., [84].

Hence, the gradient of the saturation ∇S in Equation (5.1), needs to be decomposed into one component along the streamlines and one component along lines orthogonal to the streamlines called normal lines. After this decomposition, Equation (5.1) can be written,

$$\phi \frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial \tau} + \frac{\partial}{\partial \tau}\left(\frac{g(S, \boldsymbol{x})}{\|\boldsymbol{v}\|^2}\frac{\partial S}{\partial \tau}\right) + \frac{\partial}{\partial \eta}\left(\frac{g(S, \boldsymbol{x})}{\|\boldsymbol{v}\|^2}\frac{\partial S}{\partial \eta}\right) + \frac{g(S, \boldsymbol{x})}{\|\boldsymbol{v}\|^2}\frac{\partial S}{\partial \eta}\nabla \cdot \boldsymbol{w} = q_B, \tag{5.2}$$

where we have used that $\nabla \cdot \boldsymbol{v} = 0$; furthermore, τ is the streamline coordinate defined in Equation (3.33), and η is a similar coordinate defined along the normal lines of the normal field $\boldsymbol{w}$. The normal field $\boldsymbol{w}$ is obtained by a ninety degree rotation of the flow field $\boldsymbol{v}$. We observe that last term on the left hand side of the last equation involves the divergence of the normal field, which is in general not zero even if $\boldsymbol{v}$ is divergence free. To simplify the later numerical solution procedure, we would like to get rid of this term. It turns, out that this can be done simply by a scaling of $\boldsymbol{w}$, where the scaling function can be determined analytically based on information along the normal line. It is interesting to note, that a similar type of scaling is needed in compressible streamline simulation [46], where the scaling function is the effective density along the streamlines.

Using the standard operator splitting technique [69], Equation (5.2) is then split into

$$\phi\frac{\partial S}{\partial t}+\frac{\partial f(S)}{\partial \tau}+\frac{\partial}{\partial \tau}\left(\frac{g(S,\boldsymbol{x})}{\|\boldsymbol{v}\|^2}\frac{\partial S}{\partial \tau}\right)=q_B, \qquad \text{(along streamlines)}, \tag{5.3}$$

$$\phi\frac{\partial S}{\partial t}+\frac{\partial}{\partial \tilde{\eta}}\left(\frac{g(S,\boldsymbol{x})}{\|\tilde{\boldsymbol{w}}\|^2}\frac{\partial S}{\partial \tilde{\eta}}\right)=0, \qquad \text{(along normal lines)}, \tag{5.4}$$

where $\tilde{\boldsymbol{w}}$ is the scaled normal field, and $\tilde{\eta}$ is the coordinate along $\tilde{\boldsymbol{w}}$.

The above solution procedure can be implemented as follows:

Algorithm 1 Sequential solution procedure using normal lines

```
for each global time step ΔT do
  – solve pressure equation
  – trace streamlines and normal lines
  – map saturation onto crossing points
  for each local time step Δt do
    solve parabolic 1D equation along streamlines
    solve parabolic 1D equation along normal lines
  end for
  map saturation back to pressure grid
end for
```

In the above algorithm, ΔT is the time step between pressure updates and Δt is the operator splitting time step. The innermost loop runs N times and Δt satisfies $N\Delta t = \Delta T$. Note that even smaller time steps (not shown) are used for discretizing the 1D equations within the innermost loop. An advantage of the above approach compared to a standard streamline method [55] (not using normal lines) is that we avoid additional mappings for each iteration of the innermost loop. A standard streamline method requires the cross-streamline part to be solved on the background grid, hence the solution must be mapped back and forth between the background grid for each operator splitting time step.

As noted in the beginning of this section, the method will be applied to a vertically averaged formulation, hence an extension to 3D not an issue. An extension to 3D would be far more complicated. The approach can also be used for other problems in 2D, like two-phase flow with capillary pressure, and for the advection dispersion equation in, e.g., groundwater flow.

A preliminary implementation of the procedure is described in [78].

Chapter 6

Supporting material

The thesis also includes two additional coauthored papers as supporting material. Paper G considers geostatistical generation of fractures on an aquifer analogue scale. This paper can be read as an introduction to Paper F, and to the procedure used for generation of discrete fracture networks. Paper H discusses streamline methods for simulating two-phase flow. It can be viewed as a motivation for the streamline tracing methods discussed in this thesis.

Bibliography

[1] I. Aavatsmark. An introduction to multipoint flux approxmations for quadrilateral grids. *Comput. Geosci.*, 6:405–432, 2002.

[2] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media. *Journal of Computational Physics*, 127:2–14, 1996.

[3] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. part i: Derivation of the methods. *SIAM Journal of Scientific Computing*, 19(5):1700–1716, 1998.

[4] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. part ii: Discussion and numerical results. *SIAM Journal of Scientific Computing*, 19(5):1717–1736, 1998.

[5] I. Aavatsmark, T. Barkve, and T. Mannseth. Control-volume discretization methods for 3D quadrilateral grids in inhomogeneous, anisotropic reservoirs. *SPE J.*, pages 146–154, June 1998.

[6] I. Aavatsmark and G.T. Eigestad. Numerical convergence of the MPFA O-method and U-method for general quadrilateral grids. *Int. J. Numer. Meth. Fluids*, 51:939–961, 2006.

[7] I. Aavatsmark, G.T. Eigestad, and R.A. Klausen. Numerical convergence of the MPFA O-method for general quadrilateral grids in two and three dimensions. In D.N. Arnold, P.B. Bochev, R.B. Lehoucq, R.A. Nicolaides, and M. Shashkov, editors, *Compatible spatial discretizations*, IMA volume series, pages 1–21. Springer, New York, 2006.

[8] I. Aavatsmark, G.T. Eigestad, R.A. Klausen, M.F. Wheeler, and I. Yotov. Convergence of a symmetric MPFA method on quadrilateral grids. *CG*, 11:333–345, 2007.

[9] I. Aavatsmark, G.T. Eigestad, B.T. Mallison, and J.M. Nordbotten. A compact multipoint flux approximation method with improved robustness. *Numer. Meth. Part. D.E.*, 24(5):1329–1360, 2008.

[10] P.M. Adler and J.-F. Thovert. *Fractures and fracture networks*, volume 15 of *Theory and applications of transport in porous media*. Kluwer Academic Publ., Boston, 1999.

[11] L. Agelas and R. Masson. Convergence of the finite volume MPFA O scheme for heterogeneous anisotropic diffusion problems on general meshes. *C.R. Math.*, 346:1007–1012, 2008.

[12] T. Arbogast, C.N. Dawson, P.T. Keenan, M.F. Wheeler, and I. Yotov. Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comput.*, 19(2):404–425, 1998.

[13] D.N. Arnold, D. Boffi, and R.S. Falk. Quadrilateral H(div) finite elements. *SIAM J. Numer. Anal.*, 42(6):2429–2451, 2005.

[14] A. Assteerawatt. *Flow and transport modelling of fractured aquifers based on a geostatistical approach*. PhD thesis, Dept. of hydromechanics and modeling of hydrosystems, University of Stuttgart, 2008.

[15] K. Aziz and A. Settari. *Petroleum reservoir simulation*. Elsevier Applied Science Publishers, London, 1979.

[16] B.R. Baliga and S.V. Patankar. A new finite-element formulation for convection-diffusion problems. *Numerical Heat Transfer*, 3:393–409, 1980.

[17] B.R. Baliga and S.V. Patankar. A control volume finite-element for two-dimensional fluid flow and heat transfer. *Numerical Heat Transfer*, 6:245–261, 1983.

[18] R.E. Bank and D.J. Rose. Some error estimates for the box method. *SIAM J. Numer. Anal.*, 24(4):777–787, 1987.

[19] G.I. Barenblatt, I.P. Zheltov, and I.N. Kochina. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks. *Journal of Applied Mathematics and Mechanics*, 24(5):852–864, 1960.

[20] J.A. Barker. Transport in fractured rock. In R.A. Downing and W.B. Wilkinson, editors, *Applied groundwater hydrology*, Oxford science publications, pages 199–216. Clarendon Press, Oxford, 1991.

[21] P. Bastian and R. Helmig. Efficient fully-coupled solution techniques for two-phase flow in porous media. parallel multigrid solution and large scale computations. *Adv. Water Resour.*, 23:199–216, 1999.

[22] R.P. Batycky. *A three-dimensional two-phase field scale streamline simulator. (http://www.streamsim.com/index.cfm?go=about.publications)*. PhD thesis, Stanford University, Dept. of Petroleum Engineering, 1997.

[23] J. Bear. *Dynamics of fluids in porous media*. Environmental Science Series. American Elsevier Publishing Company, New York, 1972.

[24] J. Bear. *Hydraulics of groundwater*. McGraw-Hill series in water resources and environmental engineering. McGraw-Hill, New York, 1979.

[25] J. Bear. Modeling flow and contaminant transport in fractured rocks. In J. Bear, C.-F. Tsang, and G. de Marsily, editors, *Flow and contaminant transport in fractured rock*, pages 1–37. Academic Press, San Diego, 1993.

[26] J. Bear and Y. Bachmat. *Introduction to modeling of transport phenomena in porous media*. Theory and applications of transport in porous media vol. 4. Kluwer Academic Publishers, Dordrecht, 1991.

[27] J. Bear and A. Verruijt. *Modeling groundwater flow and pollution : with computer programs for sample cases*. Theory and applications of transport in porous media ; 2. D.Reidel, Dordrecht, 1987.

[28] J. Bensabat, Q. Zhou, and J. Bear. An adaptive pathline-based particle-tracking algorithm for the eulerian-lagrangian method. *Adv. Water Resour.*, 23:383–397, 2000.

[29] B. Berkowitz. Characterizing flow and transport in fractured geological media: A review. *Adv. Water Resour.*, 25:861–884, 2002.

[30] B. Berkowitz, J. Bear, and C. Breaster. Continuum models for contaminant transport in fractured porous formations. *Water Resour. Res.*, 24(8):1225–1236, 1988.

[31] C. Bi. Superconvergence of finite volume element method for a nonlinear elliptic problem. *Numer. Meth. Part. D.E.*, 23(1):220–233, 2007.

[32] G.S. Bodvarsson, K. Pruess, and M.J. Lippmann. Modeling of geothermal systems. *Journal of Petroleum Technology*, pages 1007–1021, September 1986.

[33] F. Bratvedt, K. Bratvedt, C.F. Buchholz, T. Gimse, H. Holden, L. Holden, and N.H. Risebro. Frontline and frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking. *Surveys Math. Indust.*, 3:185–215, 1993.

[34] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer series in computational mathematics, 15. Springer-Verlag, New York, 1991.

[35] Z. Cai, J.E. Jones, S.F. McCormick, and T.F. Russell. Control-volume mixed finite element methods. *Comput. Geosci.*, 1:289–315, 1997.

[36] Z. Cai, J. Mandel, and S. McCormick. The finite volume element method for diffusion equations on general triangulations. *SIAM J. Numer. Anal.*, 28(2):392–402, 1991.

[37] Z. Cai and S. McCormick. On the accuracy of the finite volume element. *SIAM J. Numer. Anal.*, 27(3):636–655, 1990.

[38] Y. Cao, R. Helmig, and B.I. Wohlmuth. Geometrical interpretation of multi-point flux approximation l-method. *Int. J. Numer. Meth. Fluids*, (DOI: 10.1002/fld.1926), 2008.

[39] J. Carrera and L. Martinez-Landa. Mixed discrete-continuum models: A summary of experiences in test interpretation and model prediction. In B. Faybishenko, P.A. Witherspoon, and S.M. Benson, editors, *Dynamics of fluids in fractured rock*, Geophysical monograph 122, pages 251–265. American Geophysical Union, Washington, D.C., 2000.

[40] P. Castillo. A review of the local discontinuous Galerkin (LDG) method applied to elliptic problems. *Appl. Numer. Math.*, 56:1307–1313, 2006.

[41] P. Chatzipantelidis and R.D. Lazarov. Error estimates for a finite volume element method for elliptic PDEs in nonconvex polygonal domains. *SIAM J. Numer. Anal.*, 42(5):1932–1958, 2005.

[42] G. Chavent and J. Jaffré. *Mathematical models and finite elements for reservoir simulation*. North-Holland, Amsterdam, 1986.

[43] Q.-Y. Chen, J. Wan, Y. Yang, and R.T. Mifflin. Enriched multi-point flux approximation for general grids. *J. Comput. Phys.*, 227:1701–1721, 2008.

[44] Z. Chen. On the control volume finite element methods and their application to multiphase flow. *Networks and Heterogeneous Media*, 1(4):689–706, 2006.

[45] Z.-X. Chen. Transient flow of slightly compressible fluids through double-porosity, double-permeability systems – a state-of-the-art review. *Transp. Porous Media*, 4:147–184, 1989.

[46] H. Cheng, I. Osako, A. Datta-Gupta, and M.J. King. A rigorous compressible streamline formulation for two- and three-phase black-oil simulation. *SPE J.*, pages 407–417, December 2006.

[47] H.-P. Cheng, J.-R. Cheng, and G.-T. Yeh. A particle tracking technique for the lagrangian-eulerian finite element method in multi-dimensions. *Int. J. Numer. Methods Eng.*, 39:1115–1136, 1996.

[48] S.-H. Chou and Q. Li. Error estimates in L^2, H^1, and L^∞ in colvolume methods for elliptic and parabolic problems: A unified approach. *Mathematics of Computation*, 69(229):103–120, 1999.

[49] P.G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1980.

[50] B. Cockburn, J. Gopalakrishnan, and H. Wang. Locally conservative fluxes for the continuous Galerkin method. *SIAM J. Numer. Anal.*, 45(4):1742–1776, 2007.

[51] Committee on Fracture Characterization and Fluid Flow, U.S. National Commitee for Rock Mechanics, Geotechnical Board, Board on Energy and Environmental Systems, Commission on Engineering and Technical Systems, and National Research Council. *Rock fractures and fluid flow : contemporary understanding and applications*. National Academy Press, Washington, D.C., 1996.

[52] C. Cordes and W. Kinzelbach. Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. *Water Resour. Res.*, 28(11):2903–2911, 1992.

[53] M.R. Correa and A.F.D. Loula. Stabilized velocity post-processing for Darcy flow in heterogeneous porous media. *Communications in Numerical Methods in Engineering*, 23:461–489, 2007.

[54] J.-P. Croisille. Finite volume box schemes and mixed methods. *RAIRO-Math. Model. Num.*, 34(5):1087–1106, 2000.

[55] A. Datta-Gupta and M.J. King. *Streamline simulation: Theory and practice*. SPE Text book, 2007.

[56] B. Dershowitz, P. LaPointe, T. Eiben, and L. Wei. Integration of discrete feature network methods with conventional simulator approaches. *SPE Reservoir Evaluation & Engineering*, 3(2):165–170, 2000.

[57] P. Dietrich, R. Helmig, M. Sauter, H. Hötzl, J. Köngeter, and G. Teutsch, editors. *Flow and transport in fractured porous media*. Springer, Berlin, 2005.

[58] L.J. Durlofsky. Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities. *Water Resour. Res.*, 30(4):965–973, 1994.

[59] T.T. Eaton. On the importance of geological heterogeneity for flow simulation. *Sedimentary Geology*, 184:187–201, 2006.

[60] M.G. Edwards. Cross flow tensors and finite volume approximation with by deferred correction. *Comput. Methods Appl. Mech. Engrg.*, 151:143–161, 1998.

[61] M.G. Edwards. Simulation with a full-tensor coefficient velocity field recovered form a diagonal tensor solution. *SPE J.*, 5(4):387–392, 2000.

[62] M.G. Edwards. Unstructured, control-volume distributed full-tensor finite-volume schemes with flow based grids. *Comput. Geosci.*, 6:433–452, 2002.

[63] M.G. Edwards. Split full tensor discretization operators for general hexahedral grids. *SPE J.*, pages 102–108, 2004.

[64] M.G. Edwards and M. Pal. Positive-definite q-families of continuous subcell Darcy-flux CVD(MPFA) finite volume schemes and the mixed finte element method. *Int. J. Numer. Meth. Fluids*, 57:355–387, 2008.

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

[66] G.T. Eigestad. *Reservoir simulation on heterogeneous and anisotropic media with imposed flux continuity conditions for general geometries, and the inclusion of hysteresis in forward modeling*. PhD thesis, Department of mathematics, University of Bergen, 2003.

[67] G.T. Eigestad, I. Aavatsmark, and M. Espedal. Symmetry and M-matrix issues for the O-method on an unstructured grid. *Comput. Geosci.*, 6:381–404, 2002.

[68] G.T. Eigestad and R.A. Klausen. On the convergence of the multi-point flux approximation O-method: Numerical experiments for discontinuous permeability. *Numer. Meth. Part. D.E.*, 21(6):1079–1098, 2005.

[69] M. Espedal and K.H. Karlsen. Numerical solution of reservoir flow models based on large time step operator splitting algorithms. *Lecture Notes in Mathematics*, 1734:9–79, 2000.

[70] R.E. Ewing. Problems arising in the modeling of processes for hydrocarbon recovery. In R.E. Ewing, editor, *The Mathematics of reservoir simulation*, Frontiers in applied mathematics ; 1, pages 3–34. Society for Industrial and Applied Mathematics, Philadelphia, 1983.

[71] R.E. Ewing, T. Lin, and Y. Lin. On the accuracy of the finite volume element method based on piecewise linear polynomials. *SIAM J. Numer. Anal.*, 39(6):1865–1888, 2002.

[72] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of numerical analysis, vol. VII*. North-Holland, Amsterdam, 2000.

[73] S. Geiger, S. Roberts, S.K. Matthäi, C. Zoppou, and A. Burri. Combining finite element and finite volume methods for efficient multiphase flow simulations in highly heterogeneous and structurally complex geological media. *Geofluids*, 4:284–299, 2004.

[74] D.J. Goode. Particle velocity interpolation in block-centered finite difference groundwater flow models. *Water Resour. Res.*, 26(5):925–940, 1990.

[75] H. Hægland, A. Assteerawatt, H.K. Dahle, G.T. Eigestad, and R. Helmig. Comparison of cell- and vertex centered discretization methods for flow in a two-dimensional discrete fracture-matrix system. In preparation, 2008.

[76] H. Hægland, H.K. Dahle, I. Aavatsmark, G.T. Eigestad, and K.-A. Lie. Adaptive streamline tracing for streamline simulation on irregular grids. In *Proceedings of the XVI International Conference on Computational Methods in Water Resources, June, Denmark*, 2006.

[77] H. Hægland, H.K. Dahle, G.T. Eigestad, K.-A. Lie, and I. Aavatsmark. Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Adv. Water Resour.*, doi:10.1016/j.advwatres.2006.09.002, 2007.

[78] J. Hafver. Streamline methods for parabolic differential equations. Master's thesis, Department of mathematics, University of Bergen, 2008.

[79] H.H. Haldorsen and L.W. Lake. A new approach to shale management in field-scale models. *SPE J.*, pages 447–457, August 1984.

[80] Z.E. Heinemann, C.W. Brand, M. Munka, and Y.M Chen. Modeling reservoir geometry with irregular grids. *SPE Reservoir Engineering*, pages 225–232, May 1991.

[81] B. Heinrich. *Finite difference methods on irregular networks*. International series of numerical mathematics 82. Birkhäser, Basel, 1987.

[82] R. Helmig. *Multiphase flow and transport processes in the subsurface*. Environmental engineering. Springer, Berlin, 1997.

[83] R. Helmig, J. Niessner, and H. Class. Recent advances in finite element methods for multi-phase flow processes in porous media. *International Journal of Computational Fluid Dynamics*, 20(3):245–252, 2006.

[84] H. Holden, K.H. Karlsen, and K.-A. Lie. Operator splitting methods for degenerate convection-diffusion equations i: Convergence and entropy estimates. *CMS Conference Proceedings*, 29:293–316, 2000.

[85] R. Holm, R. Kaufmann, B.-O. Heimsund, E. Øian, and M.S. Espedal. Meshing of domains with complex internal geometries. *Numerical Linear Algebra with Applications*, 13(9):717–731, 2006.

[86] J. Huang and L. Li. Some superconvergence results for the covolume method for elliptic problems. *Communications in Numerical Methods in Engineering*, 17:291–302, 2001.

[87] P.S. Huyakorn and G.F. Pinder. *Computational methods in subsurface flow*. Academic Press, New York, 1983.

[88] J.C. Jaeger, N.G.W. Cook, and R.W. Zimmermann. *Fundamentals of rock mechanics*. Blackwell, Malden, Mass., 4th edition, 2007.

[89] E. Jimenez, K. Sabir, A. Datta-Gupta, and M.J. King. Spatial error and convergence in streamline simulation. *SPE Reservoir Evaluation & Engineering*, pages 221–232, June 2007.

[90] R. Juanes and S.F. Matringe. Unified formulation for high-order streamline tracing on two-dimensional unstructured grids. *Journal of Scientific Computing*, DOI10.1007/s10915-008-9228-2, 2008.

[91] E.F. Kaasschieter. Mixed finite elements for accurate particle tracking in saturated groundwater flow. *Adv. Water Resour.*, 18(5):277–294, 1995.

[92] M. Karimi-Fard, B. Gong, and L.J. Durlofsky. Generation of coarse-scale continuum flow models from detailed fracture characterizations. *Water Resour. Res.*, 42(W10423):doi:10.1029/2006WR005015, 2006.

[93] H. Kazemi and J.R. Gilman. Multiphase flow in fractured petroleum reservoirs. In J. Bear, C.-F. Tsang, and G. de Marsily, editors, *Flow and contaminant transport in fractured rock*, pages 267–323. Academic Press, San Diego, 1993.

[94] M.J. King and A. Datta-Gupta. Streamline simulation: A current perspective. *In Situ*, 22(1):91–140, 1998.

[95] P. Kipfer, F. Reck, and G. Greiner. Local exact particle tracing on unstructured grids. *Comput. Graph. Forum*, 22(2):1–9, 2003.

[96] R.A. Klausen, F.A. Radu, and G.T. Eigestad. Convergence of MPFA on triangulations and Richards' equation. *Int. J. Numer. Meth. Fluids*, (DOI: 10.1002/fld.1787), 2008.

[97] R.A. Klausen and T.F. Russell. Relationships among some locally conservative discretization methods which handle discontinuous coefficients. *Comput. Geosci.*, 8:341–377, 2004.

[98] R.A. Klausen and R. Winther. Robust convergence of multi point flux approximation on rough grids. *Numer. Math.*, 104:317–337, 2006.

[99] D. Knight and G. Mallinson. Visualizing unstructured flow data using dual stream functions. *IEEE Trans. Vis. Comput. Graph.*, 2(4):355–363, 1996.

[100] L.P. Lebedev and M.J. Cloud. *Tensor analysis*. World Scientific, River Edge, N.J., 2003.

[101] S.H. Lee, L.J. Durlofsky, M.F. Lough, and W.H. Chen. Finite difference simulation of geologically complex reservoirs with tensor permeabilities. *SPE Reservoir Evaluation & Engineering*, pages 567–574, December 1998.

[102] S.H. Lee, P. Jenny, and H.A. Tchelepi. A finite-volume method with hexahedral multiblock grids for modeling flow in porous media. *Comput. Geosci.*, 6:353–379, 2002.

[103] S.H. Lee, M.F. Lough, and C.L. Jensen. Hierarchical modeling of flow in naturally fractured formations with multiple length scales. *Water Resour. Res.*, 37(3):443–455, 2001.

[104] S.H. Lee, H.A. Tchelpi, P. Jenny, and L.J. DeChant. Implementation of a flux-continuous finite-difference method for stratigraphic, hexahedron grids. *SPE J.*, pages 267–277, September 2002.

[105] R. Li, Z. Chen, and W. Wu. *Generalized difference methods for differential equations*. Marcel Dekker, New York, 2000.

[106] N. Lu. A semianalytical method of path line computation for transient finite-difference groundwater flow models. *Water Resour. Res.*, 30(8):2449–2459, 1994.

[107] S.F. Matringe. Accurate streamline tracing and coverage. Master's thesis, Stanford university, Dept. of petr. eng., 2004.

[108] S.F. Matringe and M.G. Gerritsen. On accurate tracing of streamlines. In *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 Sep.*, 2004.

[109] S.F. Matringe, R. Juanes, and H.A. Tchelepi. Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids. *J. Comput. Phys.*, doi:10.1016/j.jcp.2006.07.004, 2006. 918.

[110] S.F. Matringe, R. Juanes, and H.A. Tchelepi. Tracing streamlines on unstructured grids form finite volume discretizations. In *Proceedings of the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, 24-27 Sept.*, 2006.

[111] S.F. Matringe, R. Juanes, and H.A. Tchelepi. Streamline tracing on general triangular and quadrilateral grids. *SPE J.*, pages 217–223, June 2007.

[112] I.D. Mishev and Q.-Y. Chen. A mixed finite volume method for elliptic problems. *Numer. Meth. Part. D.E.*, 23(5):1122–1138, 2007.

[113] J.E.P. Monteagudo and A. Firoozabadi. Control-volume model for simulation of water injection in fractured media: Incorporating matrix heterogeneity and reservoir wettability effects. *SPE J.*, pages 355–366, September 2007.

[114] R.L. Naff, T.F. Russell, and J.D. Wilson. Shape functions for velocity interpolation in general hexahedral cells. *Comput. Geosci.*, 6(3-4):285–314, 2002.

[115] T.N. Narasimhan. Multidimensional numerical simulation of fluid flow in fractured porous media. *Water Resour. Res.*, 18(4):1235–1247, 1982.

[116] T.N. Narasimhan and P.A. Witherspoon. An integrated finite difference method for analyzing fluid flow in porous media. *Water Resour. Res.*, 12(1):57–64, 1976.

[117] R.A. Nelson. Fractured reservoirs: Turning knowledge into practice. *Journal of Petroleum Technology*, pages 407–414, April 1987.

[118] R.A. Nelson. *Geologic analysis of naturally fractured reservoirs*. Gulf Professional Publ., Boston, second edition, 2001.

[119] S.P. Neuman. Trends, prospects and challenges in quantifying flow and transport through fractured rocks. *Hydrogeology Journal*, 13:124–147, 2005.

[120] J.M. Nordbotten and I. Aavatsmark. Monotonicity conditions for control volume methods on uniform parallelogram grids in homogeneous media. *Comput. Geosci.*, 9:61–72, 2005.

[121] J.M. Nordbotten, I. Aavatsmark, and G.T. Eigestad. Monotonicity of control volume methods. *Numer. Math.*, 106:255–288, 2007.

[122] J.M. Nordbotten and M.A. Celia. Similarity solutions for fluid injection into confined aquifers. *J. Fluid. Mech.*, 561:307–327, 2006.

[123] J.M. Nordbotten and G.T. Eigestad. Discretization on quadrilateral grids with improved monotonicity properties. *J. Comput. Phys.*, 203(2):744–760, 2005.

[124] J.M. Nordbotten and H. Hægland. On reproducing uniform flow exactly on general hexahedral cells using one degree of freedom per surface. *Adv. Water Resour.*, (doi:10.1016/j.advwatres.2008.11.005), 2008.

[125] A. Oliveira and A.M. Baptista. On the role of tracking on eulerian-lagrangian solutions of the transport equation. *Adv. Water Resour.*, 21:539–554, 1998.

[126] M. Pal, M.G. Edwards, and A.R. Lamb. Convergence study of a family of flux-continuous, finite-volume schemes for the general tensor pressure equation. *Int. J. Numer. Meth. Fluids*, 51:1177–1203, 2006.

[127] A. Paluszny, S.K. Matthäi, and M. Hohmeyer. Hybrid finite element-finite volume discretization of complex geologic structures and a new simulation workflow demonstrated on fractured rocks. *Geofluids*, 7:186–208, 2007.

[128] D.W. Peaceman. *Fundamentals of numerical reservoir simulation*. Developments in petroleum science. Elsevier, Amsterdam, 1977.

[129] C.F. Pinder, P.S. Huyakorn, and E.A. Sudicky. Simulation of flow and transport in fractured porous media. In J. Bear, C.-F. Tsang, and G. de Marsily, editors, *Flow and contaminant transport in fractured rock*, pages 395–435. Academic Press, San Diego, 1993.

[130] D. Pokrajec and R. Lazic. An efficient algorithm for high accuracy particle tracking in finite elements. *Adv. Water Resour.*, 25:353–369, 2002.

[131] D.W. Pollock. Semi-analytical computation of path lines for finite-difference models. *Ground Water*, 26(6):743–750, 1988.

[132] M. Prévost. *Accurate coarse reservoir modeling using unstructured grids, flow-based upscaling and streamline simulation*. PhD thesis, University of Stanford, 2003. http://geothermal.stanford.edu/pereports/search.htm.

[133] M. Prévost, M. G. Edwards, and M.J. Blunt. Streamline tracing on curvilinear structured and unstructured grids. *SPE Journal*, pages 139–148, June 2002.

[134] S. Ramadhyani and S.V. Patankar. Solution of the Poisson equation: Comparison of the Galerkin and control-volume methods. *Int. J. Numer. Methods Eng.*, 15:1395–1418, 1980.

[135] V. Reichenberger, H. Jakobs, P. Bastian, and R. Helmig. A mixed-dimensional finite volume method for two-phase flow in fractured porous media. *Adv. Water Resour.*, 29(7):1020–1036, 2006.

[136] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig, and J. Niessner. Complex gas-water process in discrete fracture-matrix systems. Project report, 2006.

[137] W. Rühaak, V. Rath, A. Wolf, and C. Clauser. 3D finite volume groundwater and heat transport modeling with non-orthogonal grids, using a coordinate transformation method. *Adv. Water Resour.*, 31:513–524, 2008.

[138] H. Rui. Symmetric modified finite volume element methods for self-adjoint elliptic and parabolic equations. *J. Comput. Appl. Math.*, 146:373–386, 2002.

[139] T.F. Russell and R.W Healy. Analytical tracking along streamlines in temporally linear raviart-thomas velocity fields. In *Proceedings of the XIII*

International Conference on Computational Methods in Water Resources, 25-29 June, Alberta, Canada, 2000.

[140] I.A. Sadarjoen, T. van Walsum, A.J.S. Hin, and F.H. Post. Particle tracing algorithms for 3d curvilinear grids. In G.M. Nielson, H. Hagen, and H. Müller, editors, *Scientific visualization: Overviews, methodologies, and techniques*, chapter 14, pages 311–335. IEEE Computer Society, Los Alamitos, Calif., 1997.

[141] M. Sahimi. Flow phenomena in rocks: from continuum models to fractals, percolation, cellular automata, and simulated annealing. *Reviews of Modern Physics*, 65(4):1393–1534, 1993.

[142] M. Sahimi. *Flow and transport in porous media and fractured rock: From classical methods to modern approaches*. VCH, Weinheim, 1995.

[143] A.L. Schafer-Perini and J.L. Wilson. Efficient and accurate front tracking for two-dimensional groundwater flow models. *Water Resour. Res.*, 27(7):1471–1485, 1991.

[144] A.E. Scheidegger. *The physics of flow through porous media*. University of Toronto Press, third edition, 1974.

[145] J.M. Shafer. Reverse pathline calculation of time-related capture zones in nonuniform flow. *Ground Water*, 25(3):283–289, 1987.

[146] M. Shashkov and S. Steinberg. *Conservative finite-difference methods on general grids*. Symbolic and numeric computation series. CRC Press, Baca Raton, 1996.

[147] Z. Sheng and G. Yuan. A nine point scheme for the approximation of diffusion operators on distorted quadrilateral meshes. *SIAM J. Sci. Comput.*, 30(3):1341–1361, 2008.

[148] S. Shirayama. Processing of computed vector fields for visualization. *J. Comput. Phys.*, 106:30–41, 1993.

[149] R. Srivastava and M.L. Brusseau. Darcy velocity computations in the finite element method for multidimensional randomly heterogeneous porous media. *Adv. Water Resour.*, 18(4):191–201, 1995.

[150] G. Strang and G.J. Fix. *An analysis of the finite element method*. Prentice-Hall series in automatic computation. Prentice-Hall, Englewood Cliffs, N.J., 1973.

[151] T. Strid, A. Rizzi, and J. Oppelstrup. Development and use of some flow visualization algorithms. *Lecture series/Von Karman Institute for Fluid Dynamics*, (7):1–56, 1989.

[152] S. Sun and M.F. Wheeler. Projections of velocity data for the compatibility with transport. *Comput. Methods Appl. Mech. Engrg.*, 195:653–673, 2006.

[153] R.V. Trujillo. *Error analysis for the finite volume element method for elliptic and parabolic partial differential equations*. PhD thesis, University of Colorado, Denver, 1996.

[154] S. Verma and K. Aziz. A control-volume scheme for flexible grids in reservoir simulation, SPE 37999. In *SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June*, 1997.

[155] J.E. Warren and P.J. Root. The behavior of naturally fractured reservoirs. *SPE J.*, pages 245–255, September 1963.

[156] M.F. Wheeler and I. Yotov. A multipoint flux mixed finite element method. *SIAM J. Numer. Anal.*, 44(5):2082–2106, 2006.

[157] S. Whitaker. Advances in theory of fluid motion in porous media. *Industrial and Engineering Chemistry*, 61(2):14–28, 1969.

[158] Y.-S. Wu, C. Haukwa, and G.S. Bodvarsson. A site-scale model for fluid and heat flow in the unsaturated zone of Yucca Mountain, Nevada. *Journal of Contaminant Hydrology*, 38(1-3):185–215, 1999.

[159] G.-T. Yeh. On the computation of Darcian velocity and mass balance in the finite element modeling of groundwater flow. *Water Resour. Res.*, 17(5):1529–1534, 1981.

[160] A. Younes and V. Fontaine. Hybrid and multi-point formulations of the loweset-order mixed methods for Darcy's flow on triangles. *Int. J. Numer. Meth. Fluids*, 58:1041–1062, 2008.

[161] D. Zhang. *Stochastic methods for flow in porous media: coping with uncertainties*. Academic Press, San Diego, Calif., 2002.

[162] C. Zheng and G.D. Bennett. *Applied contaminant transport modeling*. Wiley-Interscience, New York, second edition, 2002.

[163] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The finite element method : its basis and fundamentals*. Elsevier, Amsterdam, 6th edition, 2005.

Part II

Included Papers

Paper A

Improved streamlines and time-of-flight for streamline simulation on irregular grids

*

* In *Advances in Water Resources*, 30(4), 1027-1045, 2007

Available online at www.sciencedirect.com
ScienceDirect
Advances in Water Resources 30 (2007) 1027–1045
Advances in Water Resources
www.elsevier.com/locate/advwatres

Improved streamlines and time-of-flight for streamline simulation on irregular grids

H. Hægland [a,*], H.K. Dahle [a], G.T. Eigestad [a,b], K.-A. Lie [c], I. Aavatsmark [b]

[a] Department of Applied Mathematics, University of Bergen, 5008 Bergen, Norway
[b] Center for Integrated Petroleum Research, University of Bergen, 5020 Bergen, Norway
[c] SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, No-0314 Oslo, Norway

Received 30 September 2005; received in revised form 18 September 2006; accepted 21 September 2006
Available online 13 November 2006

Abstract

Streamline methods have shown to be effective for reservoir simulation. For a regular grid, it is common to use the semi-analytical Pollock's method to obtain streamlines and time-of-flight coordinates (TOF). The usual way of handling irregular grids is by trilinear transformation of each grid cell to a unit cube together with a linear flux interpolation scaled by the Jacobian. The flux interpolation allows for fast integration of streamlines, but is inaccurate even for uniform flow. To improve the tracing accuracy, we introduce a new interpolation method, which we call *corner-velocity interpolation*. Instead of interpolating the velocity field based on discrete fluxes at cell edges, the new method interpolates directly from reconstructed point velocities given at the corner points in the grid. This allows for reproduction of uniform flow, and eliminates the influence of cell geometries on the velocity field. Using several numerical examples, we demonstrate that the new method is more accurate than the standard tracing methods.

Keywords: Streamline tracing; Time-of-flight; Irregular grids

1. Introduction

Streamlines, pathlines, and streaklines are convenient tools for describing and visualizing flow given by an external velocity field $\boldsymbol{q} = (q_x, q_y, q_z)$. *Streamlines* are a family of curves $\boldsymbol{s}(\tau)$ that are instantaneously tangent to the velocity vector $\boldsymbol{q}$ at every point

$$\frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}\tau} = \boldsymbol{q}.$$

Streamlines can be traced for any vector field, although the most common is that $\boldsymbol{q}$ represents a velocity obtained from the solution of a set of flow equations. For incompressible flow, streamlines defined at a single instant do not intersect and cannot begin or end inside the fluid. *Streamtubes* are regions bounded by streamlines. Because streamlines are tangent to the velocity field, fluid that is inside a streamtube must remain forever within the same streamtube.

A *pathline* $\boldsymbol{x}(t)$ is the trajectory traced out by an imaginary massless particle following the flow of the fluid from a given starting point,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{q}(\boldsymbol{x}, t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{1}$$

A *streakline* is the locus at a given instance of the positions of all fluid particles that have gone through a fixed spatial point in the past. In steady flow streamlines, streaklines, and pathlines coincide; in unsteady flow they can be different.

The integration of (1) to obtain particle paths and/or travel times, is known as *particle tracking*, for which there exists a rich literature. The particle tracking literature is primarily concerned with problems where the velocity field is only known at a finite set of points, either measured or calculated from a flow model, and interpolation is needed to

* Corresponding author. Tel.: +47 55 584865; fax: +47 55 589672.
E-mail address: hakonh@mi.uib.no (H. Hægland).

0309-1708/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2006.09.002

integrate pathlines. In computational fluid dynamics, particle tracking has been used for visualization [30,32,48,53,54]. Velocity interpolation in control-volume mixed finite-element methods is a related subject to particle tracking and has been considered in [40]. Within groundwater flow simulation, particle tracking is used to model contaminant transport [4,13,42,47,49,50]. In visualization, the integration of (1) is usually done numerically using a Runge–Kutta type solver, whereas in groundwater flow, semi-analytical integration is the most common.

In the following we regard *streamline tracing* as a subset of particle tracking, since streamlines may be computed by particle tracking if we introduce the streamline parameter τ as an artificial time variable for which the instantaneous velocity field $\boldsymbol{q}(\boldsymbol{x}, t)$ is steady. In this paper we consider streamline tracing in the context of streamline simulation of flow in hydrocarbon reservoirs [2,5,29]. In this case, the fluid velocity $\boldsymbol{q}$ is typically given as the numerical solution of a set of flow equations for $\boldsymbol{q}$ and the fluid pressure p of the form

$$cp_t + \nabla \cdot \boldsymbol{q} = b, \quad \boldsymbol{q} = -\boldsymbol{a}(\boldsymbol{x})\nabla p.$$

The two equations are commonly referred to as the pressure equation and Darcy's law, respectively. How the corresponding discrete velocity approximation is defined, depends on the numerical method:

- For finite-difference methods, the pressure is usually computed at cell centers, and fluxes can be obtained at cell edges by application of a discrete form of Darcy's law [60].
- For finite-element methods, the numerical solution gives a continuously defined pressure approximation given as the sum of the basis functions for all elements weighted by the corresponding node values. Although a continuously defined velocity can be obtained from Darcy's law, a better strategy is given in [13,17], where continuous fluxes are obtained at cell edges.
- Mixed finite-element methods *solve* for velocity and pressure simultaneously, resulting in a more accurate velocity field than for finite differences and standard finite elements. The continuously defined velocity is given by the degrees of freedom at the edges and the corresponding basis functions [17,28]; see also [27,38,39].
- Finite-volume methods include multi-point flux approximations [1,18] and control-volume finite-element methods [8,59]. In these methods fluxes are computed at cell edges.

In other words, a continuously defined velocity field is obtained only for the mixed finite-element method. For the other methods one must use an interpolation scheme to determine the velocity from the discrete fluxes at the cell edges.

In reservoir simulation and groundwater flow, the predominant way of computing streamlines is by use of a semi-analytical technique. In semi-analytical methods [22,35,44,49], the interpolation of the velocity is simple enough that analytical integration is possible within each grid cell. As an example, let us consider the popular method of Pollock [44]. Given an entry point of a streamline into a grid cell, Pollock's method starts by mapping the grid cell onto the unit square (or unit cube in 3D). Each component of the velocity field is then approximated in reference space by a linear function, in which case the streamline path in each direction is given as an exponential function of the travel time. To trace the streamline, Pollock's method determines the travel time through the grid block as the minimum time to exit in each spatial direction, which is given by a logarithmic expression. Then the travel time is used to compute the exit point and the exit point is mapped back into physical space to give the entry point into the next cell, and so on. In groundwater flow and visualization, more complicated interpolation schemes have been used, where numerical integration is needed, using Euler's method or higher-order Runge–Kutta methods [4,10,42,43,50].

Tracing of streamlines for use in flow simulations has been investigated in [25,27,29,36–39,45,46]. In the current paper we focus on irregular grids in three spatial dimensions consisting of hexahedral grid cells with curved surfaces. Streamline tracing may then be performed by a method due to Prévost et al. [46]. This method is a simplification of work done by Cordes and Kinzelbach [13], where Pollock's method is extended to irregular grids. Each grid cell in physical space $\mathscr{P}$ is transformed to a unit cube in a reference space $\mathscr{R}$ using a standard isoparametric trilinear transformation [12]. Next, the velocity in $\mathscr{R}$ is approximated by a linear flux interpolation scaled by an approximation of the Jacobian determinant of the transformation. Finally, the streamline segments passing through each cell can be integrated in $\mathscr{R}$ and mapped back to $\mathscr{P}$. Henceforth, we will call this method *standard flux-mapping*, SFM.

As we shall see, the standard flux mapping cannot reproduce a uniform flow field on irregular hexahedrons. Uniform flow or almost uniform flow are important cases to consider, since such flow patterns are likely to occur in large parts of a hydrocarbon reservoir. For example, if the driving force for the flow in the reservoir is gravity, and the permeability of the medium is almost homogeneous, the flow locally is almost uniform in the z-direction of the reservoir. Reproduction of uniform flow is important if the grid cells are small compared to the variation of the velocity. Also, the failure to reproduce uniform flow will produce errors in the interpolated velocity for the SFM even for nonuniform flow. The error will increase with the irregularity of the grid cells.

To handle the difficulty associated with the standard flux mapping, we consider an alternative method, which we call *corner-velocity interpolation* (CVI). Instead of interpolating the velocity field based on discrete fluxes at cell edges, we interpolate directly from point velocities given at the corner points in the grid. This allows for reproduction of uniform flow, and eliminates the influence of the cell geometry on the velocity field. In streamline simulation, we usually only

know fluxes, so we also present a method for reconstructing the corner-point velocities in each grid cell from fluxes.

A radically different strategy was recently introduced by Matringe et al. [38,39] and Juanes and Matringe [27] for handling irregular grids consisting of triangles and quadrilaterals in 2D. They propose both low and high-order tracing methods based on the mathematical framework of mixed finite-element methods and the associated velocity spaces.

The rest of this paper is organized as follows: First we give a short introduction to streamline simulation. Then Section 3 describes the standard method for streamline tracing on irregular grids. We start with a brief review of Pollock's method [44] for streamline tracing on a Cartesian grid. Then a velocity transformation from physical to reference space is described. Considering this velocity transformation we discuss the methods of Cordes and Kinzelbach [13] and Prévost et al. [46] for streamline tracing on irregular grids. A recent improvement of this standard flux mapping due to Jimenez et al. [25] is also discussed. The latter method is here denoted by *extended flux-mapping* (EFM). Finally, the extension of these methods to 3D will motivate the corner-velocity interpolation method (CVI).

In Section 4, the CVI method is described first for the 2D case and then extended to 3D. We use bilinear or trilinear interpolation of the velocities at the corners of the cell [9,54]; an important part of the method is the reconstruction of the corner velocities from the given fluxes.

Finally, in Section 5 the three methods (SFM, EFM, and CVI) are compared by numerical experiments for different grids and flow fields in 3D. We consider uniformly and nonuniformly refined grids, including random hexahedral, pyramidal, and a simplistic real field grid. The flow fields may be analytical solutions of the pressure equation, including uniform, combined uniform and nonuniform, and pure nonuniform flow; or, realistic flow fields where analytical solutions are not available. Further, the CVI method is also tested for use with half-edge fluxes in 2D.

2. Background: streamline simulation

Multiphase flow in porous media is usually modeled by a coupled set of differential equations. Using the so-called fractional flow formulation, the flow of e.g., two phases can be described by a parabolic equation for the fluid pressure p (neglecting for simplicity gravity and capillary forces)

$$c_t \partial_t p + \nabla \cdot \boldsymbol{q} = b_p, \tag{2}$$

where $\boldsymbol{q}$ is the total velocity (sum of phase velocities), c_t is the total compressibility, and b_p is a source term. Eq. (2) is linked to a transport equation for the fluid saturation S

$$\phi \partial_t S + \nabla \cdot (\boldsymbol{q} f(S, \boldsymbol{x})) = b_s, \tag{3}$$

through Darcy's equation for the velocity,

$$\boldsymbol{q} = -\lambda(S, \boldsymbol{x}) \nabla p. \tag{4}$$

Here, ϕ, λ, f, and b_s denote porosity, total mobility, fractional flow function, and source terms, respectively.

The basis for any streamline simulation method is a sequential splitting of the coupled pressure and saturation equations, in which one first fixes the saturation and solves the pressure Eq. (2) and Darcy's law (4). The pressure and velocity fields are then used as parameters while advancing the transport equation (3) a given time step. Finally, the new saturation field is used as input parameter for a new pressure solution step, and so on.

In reservoir simulation, the streamline parameter τ is called time-of-flight, since it can be interpreted as the travel time of a neutral particle along the streamline. Together with the bi-stream functions ψ and χ, for which $\rho \boldsymbol{q} = \nabla\psi \times \nabla\chi$, the time-of-flight τ forms an alternative set of coordinates for three-dimensional space [3,11,29]. Here, the effective density ρ reduces to $\rho \equiv 1$ for incompressible flows, see [11]. The Jacobian of the transformation from physical coordinates (x, y, z) to time-of-flight coordinates (τ, ψ, χ) simply equals ϕ. Using this, and the fact that $\boldsymbol{q}$ is orthogonal to $\nabla\psi$ and $\nabla\chi$, allows us to simplify the directional gradient along $\boldsymbol{q}$ as follows:

$$\boldsymbol{q} \cdot \nabla = \phi \frac{\partial}{\partial \tau}.$$

This operator identity is a key point in any streamline method, allowing the multidimensional transport equation (3) to be transformed to a family of one-dimensional transport equations along streamlines (which are straight lines in (τ, ψ, χ) space),

$$\partial_t S + \partial_\tau f(S) = b_s - f(S) \nabla \cdot \boldsymbol{q}. \tag{5}$$

The last term on the right-hand side accounts for compression or expansion of fluids in the case of compressible flows. For incompressible flows, $\nabla \cdot \boldsymbol{q} = 0$ outside wells. Solving the family of one-dimensional problems (5) on a representative set of streamlines is often much faster than solving (3) over a grid in physical space.

Streamline simulation has grown in popularity in the last years due to its ability to deliver fast and accurate simulation of large reservoir models using simplified flow physics [29,55]. However, current streamline simulators are also capable of including gravity and capillary effects by the means of operator splitting [6,20,21] and simulating complex flow physics like miscibility [26,58], three-phase [24,34] and compositional flow [14,56], and dual-porosity models [15,57].

3. Streamline tracing on irregular grids

In this section we will describe the standard methods that are used in current commercial streamline codes for tracing streamlines on irregular grids. The method relies on a trilinear mapping from physical space to a reference space, linear interpolation of each velocity component, and analytical solution of the streamline equation (1) within each grid cell. We will then show that the method, and its recent extension due to Jimenez et al. [25], are unable to correctly reproduce uniform flow on arbitrary irregular grids in three spatial dimensions.

3.1. Introduction: tracing on cartesian grids

To motivate the description of streamline tracing on irregular grids, we start by discussing the basic version on Cartesian grids, which is commonly referred to as Pollock's method in the literature. As we saw in the introduction, Pollock's method builds a streamline as a series of (small) line segments that each cross a grid cell in physical space. The segments are constructed such that the exit point of the streamline in one cell is the entrance point in the next cell. For the development in this paper, it is sufficient to consider the method in the unit square (or unit cube in 3D).

3.1.1. Pollock's method

The method will be presented for 2D; the extension to 3D should be obvious. Linear interpolation of the edge fluxes is then used to define a velocity field (see Fig. 1)

$$\boldsymbol{q}^{\mathrm{I}}(x,y) \equiv \begin{bmatrix} F_{x0}(1-x)+F_{x1}x \\ F_{y0}(1-y)+F_{y1}y \end{bmatrix}, \quad 0 \leqslant x \leqslant 1,\ 0 \leqslant y \leqslant 1. \tag{6}$$

Here the superscript I refers to the fact that the velocity field is interpolated based on fluxes given at the edges, as shown in Fig. 1.

Having defined a velocity field, the streamline $\boldsymbol{s}(t) = [x(t), y(t)]$ is found by integrating the system of ODEs in (1):

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} = q_x^{\mathrm{I}}(x), & x(0) = x_0, \\ \frac{\mathrm{d}y}{\mathrm{d}t} = q_y^{\mathrm{I}}(y), & y(0) = y_0, \end{cases} \tag{7}$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$. Since q_x^{I} depends only on x, and q_y^{I} depends only on y, the streamline can be found analytically [44]: Assuming $F_{x0} \neq F_{x1}$ and $F_{y0} \neq F_{y1}$, integration of each of the equations in (7) yields two separate expressions for the travel time along the streamline as a function of x and y, respectively,

$$t_x = \frac{1}{F_{x1}-F_{x0}} \ln\left(\frac{q_x^{\mathrm{I}}(x)}{q_x^{\mathrm{I}}(x_0)}\right), \tag{8}$$

$$t_y = \frac{1}{F_{y1}-F_{y0}} \ln\left(\frac{q_y^{\mathrm{I}}(y)}{q_y^{\mathrm{I}}(y_0)}\right). \tag{9}$$

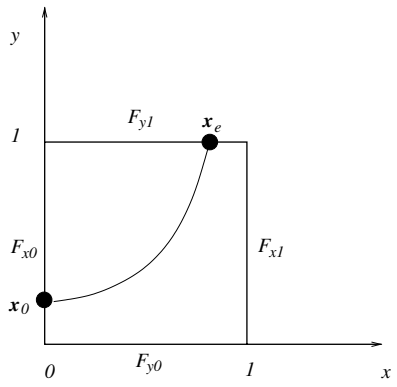


Fig. 1. Streamline tracing on a unit square.

By inserting $x = 0$ and $x = 1$ in (8) and $y = 0$ and $y = 1$ in (9), we determine the times t_{x0}, t_{x1}, t_{y0}, and t_{y1}, respectively, when the streamline crosses the corresponding straight lines. (Notice that these times may be negative or infinite.) From these four travel times we can easily determine the exit time t_e when the streamline leaves the unit square. The exit point $\boldsymbol{x}_e$ is then found by inserting t_e in (8) and (9).

3.1.2. Applying Pollock's method in reservoir simulation

For flow in porous media, the flow velocity $\boldsymbol{q}$ is obtained by solving (2) and (4) using e.g., a flux continuous scheme [1] to provide fluxes on each grid cell edge. In order to obtain particle velocities, these fluxes should be divided by the porosity. We assume for the moment that the grid cells can be any quadrilateral. The fluxes will be used to define a velocity field $\boldsymbol{q}^{\mathrm{I}}$ that approximates $\boldsymbol{q}$.

The flux is the integral of the normal component of the velocity field across an edge. We will require that $\boldsymbol{q}^{\mathrm{I}}$ reproduces the given edge fluxes,

$$F_E = \int_E \boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{v}_E \, \mathrm{d}s, \tag{10}$$

where the subscript E refers to the edge, F_E is the flux over the edge, and $\boldsymbol{v}_E$ is a unit normal to the edge. At the common edge between two adjacent grid cells, the absolute value of the flux is the same seen from both cells. By defining $\boldsymbol{v}_E$ suitably, we can assure that the sign of the flux is also the same. Thus only one flux is needed per cell edge.

We next assume that the normal component $(\boldsymbol{q}^{\mathrm{I}} \cdot \boldsymbol{v}_E)$ is constant along a given edge. Then (10) becomes

$$F_{E_i} = (\boldsymbol{q}^{\mathrm{I}}|_{E_i} \cdot \boldsymbol{v}_{E_i})\,|E_i|, \quad i = 1,2,3,4, \tag{11}$$

where E_i is one of the four edges, $\boldsymbol{q}^{\mathrm{I}}|_{E_i}$ denotes $\boldsymbol{q}^{\mathrm{I}}$ evaluated at a point on the edge, and $|E_i|$ is the length of the edge. Using this, combined with (10) for a unit square, gives

$$q_x^{\mathrm{I}}(0,y) = F_{x0}, \quad q_y^{\mathrm{I}}(x,0) = F_{y0},$$
$$q_x^{\mathrm{I}}(1,y) = F_{x1}, \quad q_y^{\mathrm{I}}(x,1) = F_{y1},$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$, respectively. We are now in a position to introduce the velocity interpolation (6) and use Pollock's method to obtain the streamlines, as described above.

Finally, note that the interpolation step is not always necessary since (6) can be obtained directly when solving the pressure equation with a mixed finite-element method using the lowest-order Raviart-Thomas (RT_0) elements [7].

3.2. Streamline and velocity in curvilinear coordinates

Streamline tracing on irregular grid cells involves introducing a curvilinear coordinate system [33] for each grid cell. In this section we describe the coordinate transformation and how to express the velocity in curvilinear coordinates. This transformation of coordinates and velocity will be fundamental for the development of the CVI method. Additionally, it will be used to derive the SFM and EFM methods.

Consider a quadrilateral grid cell in physical space $\mathscr{P}$ given by the four corner points $\boldsymbol{x}_i = [x_i, y_i]$, $i = 1, \ldots, 4$. By using the bilinear isoparametric transformation [12,13,46]

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) \equiv \sum_{i=1}^{4} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}), \tag{12}$$

each grid cell is transformed into a unit square in the reference space $\mathscr{R}$. Here $\hat{\boldsymbol{x}} = [\hat{x}, \hat{y}]$ is a point in $\mathscr{R}$; $\boldsymbol{x}(\hat{\boldsymbol{x}}) = [x(\hat{x}, \hat{y}), y(\hat{x}, \hat{y})]$ is a point in physical space $\mathscr{P}$; and $\phi_i(\hat{x}, \hat{y})$, $i = 1, \ldots, 4$, are the standard bilinear shape functions on the unit square.

Later we will compute normal vectors to the cell edges. In order to obtain a well-defined direction of these normals we require that $\boldsymbol{x}_i$, $i = 1, \ldots, 4$, are the logically bottom-left, bottom-right, top-left, and top-right corner of the quadrilateral, respectively.

3.2.1. A velocity transformation

If we can describe the velocity in bilinear coordinates, the streamline can be integrated in bilinear coordinates, and since each grid cell is a unit square in $\mathscr{R}$, Pollock's method is applicable. To obtain the streamline $\boldsymbol{s}(t)$ in $\mathscr{P}$, the bilinear transformation is applied to the streamline $\hat{\boldsymbol{s}}(t)$ in $\mathscr{R}$; see Fig. 2. From this we can use the chain rule to deduce the velocity in $\mathscr{R}$,

$$\boldsymbol{q} \equiv \frac{\mathrm{d}\boldsymbol{s}}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{x}(\hat{\boldsymbol{s}}(t))}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\hat{\boldsymbol{x}}} \frac{\mathrm{d}\hat{\boldsymbol{s}}}{\mathrm{d}t} = \boldsymbol{J}\hat{\boldsymbol{q}}. \tag{13}$$

Here $\hat{\boldsymbol{q}} = \mathrm{d}\hat{\boldsymbol{s}}/\mathrm{d}t$ is the velocity in $\mathscr{R}$, $\boldsymbol{q}$ is the velocity in $\mathscr{P}$, and $\boldsymbol{J} = \mathrm{d}\boldsymbol{x}/\mathrm{d}\hat{\boldsymbol{x}}$ is the Jacobian matrix of the transformation. Thus, the transformed velocity is given by

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q}. \tag{14}$$

Note that the Piola transformation [7,27] of a vector field given by

$$\hat{\boldsymbol{q}}^{\boldsymbol{P}} = (\det \boldsymbol{J})\boldsymbol{J}^{-1}\boldsymbol{q} \tag{15}$$

is constructed so that fluxes are preserved in reference space. By comparing (14) and (15) we see that $\hat{\boldsymbol{q}}^{\boldsymbol{P}} = (\det \boldsymbol{J})\hat{\boldsymbol{q}}$.

3.3. Extending Pollock's method to irregular grids

Tracing on irregular grids is done in reference space $\mathscr{R}$ using (14). The Jacobian matrix is given by

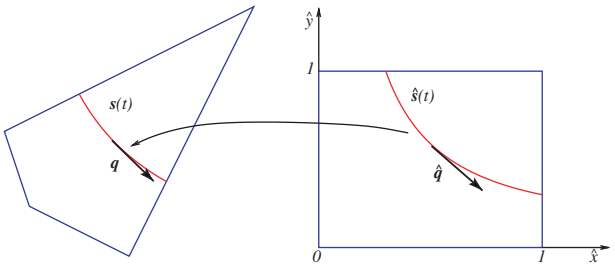


Fig. 2. Transformation of a streamline and velocity from reference space $\mathscr{R}$ to physical space $\mathscr{P}$.

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{bmatrix} = [\,\boldsymbol{u}_x \quad \boldsymbol{u}_y\,], \tag{16}$$

where $\boldsymbol{u}_x$ and $\boldsymbol{u}_y$ are the base vectors of the bilinear coordinates. Thus, the inverse of the Jacobian matrix can be expressed in terms of contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, as

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \end{bmatrix}, \tag{17}$$

where

$$\boldsymbol{n}_x = [\partial y/\partial \hat{y}, -\partial x/\partial \hat{y}]^{\mathrm{T}}, \quad \text{and}$$
$$\boldsymbol{n}_y = [-\partial y/\partial \hat{x}, \partial x/\partial \hat{x}]^{\mathrm{T}}. \tag{18}$$

The contravariant vectors, $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$, are normal vectors to edges of the quadrilateral in physical space $\mathscr{P}$. These vectors are generally not constant, but when evaluated at a particular edge of the quadrilateral, they are constant; and the length of these vectors is then equal to the length of the edge. It follows from (14) and (17) that,

$$\hat{\boldsymbol{q}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q} \cdot \boldsymbol{n}_x \\ \boldsymbol{q} \cdot \boldsymbol{n}_y \end{bmatrix}. \tag{19}$$

Next, we approximate $\boldsymbol{q}$ and $\hat{\boldsymbol{q}}$ by $\boldsymbol{q}^{\mathrm{I}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}$, respectively, based on the given fluxes in $\mathscr{P}$. Still using (11) for $\boldsymbol{q}^{\mathrm{I}}$, and recognizing the dot products in (19) as fluxes since $\boldsymbol{n}_E = \boldsymbol{v}_E|E|$, we can define $\hat{\boldsymbol{q}}^{\mathrm{I}}$ by the use of a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}. \tag{20}$$

This expression is the basis for the standard flux-mapping (SFM) method [46] and the extended flux-mapping (EFM) method of Jimenez et al. [25]. In the SFM method [46], the bilinear Jacobian determinant is replaced by a constant value in order to make analytical integration possible. Hence,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{SFM}} = \frac{1}{\det \boldsymbol{J}_m} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}, \tag{21}$$

where $\boldsymbol{J}_m = \boldsymbol{J}(0.5, 0.5)$ is the Jacobian matrix evaluated at the midpoint of the reference element. In [40] it was shown that (20) is exact for uniform flow in 2D. Since the Jacobian determinant only scales the absolute value of the velocity in (21), SFM reproduces the shape of the streamlines exactly for uniform flow. However, approximating the Jacobian determinant by a constant introduces errors in computing time-of-flight, as noted by Jimenez et al. [25]. They demonstrated that by using a pseudo time-of-flight τ, the velocity in (20) can be integrated analytically by rewriting (20) as

$$\begin{cases} \dfrac{\mathrm{d}\hat{x}}{F_{x0}(1-\hat{x}) + F_{x1}\hat{x}} = \dfrac{\mathrm{d}t}{\det \boldsymbol{J}} = \mathrm{d}\tau, \\ \dfrac{\mathrm{d}\hat{y}}{F_{y0}(1-\hat{y}) + F_{y1}\hat{y}} = \dfrac{\mathrm{d}t}{\det \boldsymbol{J}} = \mathrm{d}\tau. \end{cases} \tag{22}$$

Then Pollock's method is used to find $\hat{x}(\tau)$ and $\hat{y}(\tau)$ and the exit pseudo time τ_e. To find the real exit time, t_e, we integrate the determinant of the Jacobian

$$t_e = \int_0^{t(\tau_e)} \mathrm{d}t = \int_0^{\tau_e} \det \boldsymbol{J}(\hat{x}(\tau), \hat{y}(\tau))\, \mathrm{d}\tau. \tag{23}$$

Thus, the EFM method is characterized by the velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}}$ given in (20), where the Jacobian determinant is evaluated exactly.

3.4. Extending the SFM and EFM methods to 3D

We consider an irregular grid consisting of hexahedral grid cells with fluxes computed on the faces of each cell. Each hexahedron in physical space $\mathscr{P}$ will be defined as a one-to-one trilinear map [12,13,46] of a unit cube in a reference space $\mathscr{R}$. The trilinear map is given by

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) = \sum_{i=1}^{8} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}, \hat{z}), \tag{24}$$

where $\boldsymbol{x}_i$, $i = 1, \ldots, 8$ are the coordinates of the eight corners defining the grid cell, and $\phi_i(\hat{x}, \hat{y}, \hat{z})$, $i = 1, \ldots, 8$ are the standard trilinear shape functions on the unit cube. Note that these hexahedrons generally have curved surfaces.

We remark that the mapping gives a natural definition of the faces of the hexahedron in physical space. We define general x-, y-, and z-surfaces in $\mathscr{P}$ by

$$\begin{aligned} s_x(\hat{x}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \leqslant \hat{y} \leqslant 1, 0 \leqslant \hat{z} \leqslant 1\}, \\ s_y(\hat{y}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \leqslant \hat{x} \leqslant 1, 0 \leqslant \hat{z} \leqslant 1\}, \\ s_z(\hat{z}) &= \{\boldsymbol{x}(\hat{x}, \hat{y}, \hat{z}) : 0 \leqslant \hat{x} \leqslant 1, 0 \leqslant \hat{y} \leqslant 1\}. \end{aligned} \tag{25}$$

For the primary faces, $s_x(0)$, $s_x(1)$, etc., we will also use the notation S_{x0}, S_{x1}, respectively.

The velocity in physical space $\mathscr{P}$ is related to the velocity in reference space $\mathscr{R}$ by (13). The Jacobian matrix of the transformation is written,

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} & \frac{\partial x}{\partial \hat{z}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} & \frac{\partial y}{\partial \hat{z}} \\ \frac{\partial z}{\partial \hat{x}} & \frac{\partial z}{\partial \hat{y}} & \frac{\partial z}{\partial \hat{z}} \end{bmatrix} = [\, \boldsymbol{u}_x \quad \boldsymbol{u}_y \quad \boldsymbol{u}_z \,]. \tag{26}$$

Here $\boldsymbol{u}_x$, $\boldsymbol{u}_y$, and $\boldsymbol{u}_z$ are the covariant base vectors of the trilinear coordinates. It follows that the inverse of $\boldsymbol{J}$ can be expressed in terms of contravariant vectors $\boldsymbol{n}_x$, $\boldsymbol{n}_y$, and $\boldsymbol{n}_z$ such that

$$\boldsymbol{J}^{-1} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{n}_x^{\mathrm{T}} \\ \boldsymbol{n}_y^{\mathrm{T}} \\ \boldsymbol{n}_z^{\mathrm{T}} \end{bmatrix}, \tag{27}$$

where

$$\boldsymbol{n}_x = \boldsymbol{u}_y \times \boldsymbol{u}_z, \quad \boldsymbol{n}_y = \boldsymbol{u}_z \times \boldsymbol{u}_x, \quad \boldsymbol{n}_z = \boldsymbol{u}_x \times \boldsymbol{u}_y. \tag{28}$$

Therefore,

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} \boldsymbol{q} \cdot \boldsymbol{n}_x \\ \boldsymbol{q} \cdot \boldsymbol{n}_y \\ \boldsymbol{q} \cdot \boldsymbol{n}_z \end{bmatrix}. \tag{29}$$

As in the 2D case, the EFM method of Jimenez et al. [25] is based on (29) in combination with a linear flux interpolation,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \\ F_{z0}(1-\hat{z}) + F_{z1}\hat{z} \end{bmatrix}, \tag{30}$$

where F_{x0} is the given flux over the face S_{x0} in $\mathscr{P}$ corresponding to $\hat{x}$=0, etc. We will later use the normal vectors at the six primary faces defined as,

$$\boldsymbol{n}_{x0}(\hat{y}, \hat{z}) \equiv \boldsymbol{n}_x(0, \hat{y}, \hat{z}), \quad \boldsymbol{n}_{x1}(\hat{y}, \hat{z}) \equiv \boldsymbol{n}_x(1, \hat{y}, \hat{z}), \tag{31}$$

$$\boldsymbol{n}_{y0}(\hat{x}, \hat{z}) \equiv \boldsymbol{n}_y(\hat{x}, 0, \hat{z}), \quad \boldsymbol{n}_{y1}(\hat{x}, \hat{z}) \equiv \boldsymbol{n}_y(\hat{x}, 1, \hat{z}), \tag{32}$$

$$\boldsymbol{n}_{z0}(\hat{x}, \hat{y}) \equiv \boldsymbol{n}_z(\hat{x}, \hat{y}, 0), \quad \boldsymbol{n}_{z1}(\hat{x}, \hat{y}) \equiv \boldsymbol{n}_z(\hat{x}, \hat{y}, 1), \tag{33}$$

where $(\hat{x}, \hat{y}, \hat{z}) \in [0,1] \times [0,1] \times [0,1]$.

In the SFM method by Prévost et al. [46], the Jacobian in (30) is evaluated at the midpoint of the unit cube.

3.5. Reproduction of uniform flow

By uniform flow we refer to flow given by a constant velocity field $\boldsymbol{q}$. Obviously, uniform flow leads to straight and parallel streamlines, and therefore the time-of-flight is equal at all points having the same distance from the inflow boundary.

In 2D, the linear flux interpolation used by EFM is exact for uniform flow, as shown in [40]. However, this is not the case in 3D: in [40] it was shown that the flux of a uniform flow field will vary quadratically. This might lead one to believe that replacing the linear interpolation in (30) with a quadratic, would solve the problem.

The difficulty with any flux interpolation can be seen from the following argument: Consider for instance the surface S_{x0}. A normal vector $\boldsymbol{n}_{x0}$ to this face at $\boldsymbol{x}(0, \hat{y}, \hat{z})$ is given by (31). The absolute value of $\boldsymbol{n}_{x0}$ equals the surface Jacobian, which in the case of planar faces is constant only for parallelograms. For nonplanar faces the direction of the normal vector is not constant either. By inserting $\hat{x} = 0$ in (29) and (30) we see that both SFM and EFM use the following approximation:

$$\boldsymbol{q} \cdot \boldsymbol{n}_{x0} = F_{x0}. \tag{34}$$

For uniform flow, $\boldsymbol{q}$ is constant but $\boldsymbol{n}_{x0}$ will generally not be a constant, as noted above. The normal vector in (34) cannot change at a fixed point, so in effect we will trace the streamline using a velocity $\tilde{\boldsymbol{q}}$ such that $\tilde{\boldsymbol{q}} \cdot \boldsymbol{n}_{x0} = F_{x0}$, and $\tilde{\boldsymbol{q}}$ compensates for the fact that $\boldsymbol{n}_{x0}$ is not constant. Hence, the tracing velocity $\tilde{\boldsymbol{q}}$ will depend on the normal vector $\boldsymbol{n}_{x0}$, or in other words, depend on the geometry of the cell. Therefore uniform flow cannot be reproduced.

4. Corner-velocity interpolation

In the previous section we described how the SFM and the EFM methods fail to reproduce uniform flow on e.g., grids with nonplanar faces. To remedy this problem, we

will propose a different velocity interpolation scheme, which we will denote *corner-velocity interpolation* (CVI). For simplicity, the method will first be introduced in 2D and then extended to 3D in Section 4.2.

4.1. Interpolation in 2D

As for the SFM and EFM methods introduced in Section 3.3, we will formulate the CVI method using a cell-by-cell integration on the unit cube in reference space $\mathscr{R}$. However, the method may also be used to integrate streamlines directly in physical space; see [23] for more details. To formulate the CVI method, we start with the relation (19) for the velocity in $\mathscr{R}$, where the unknown quantity is the velocity $\boldsymbol{q}$ in $\mathscr{P}$. We approximate $\boldsymbol{q}$ by a bilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1,\ldots,4$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(\boldsymbol{x}(\hat{x},\hat{y})) \equiv \sum_{i=1}^{4} \boldsymbol{q}_i \phi_i(\hat{x},\hat{y}). \tag{35}$$

The corner velocities $\boldsymbol{q}_i$ will be reconstructed from the given fluxes, such that $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ is exact for uniform flow. This means that all $\boldsymbol{q}_i$ will be equal if $\boldsymbol{q}$ is constant.

Hence the CVI method is characterized by the following velocity interpolation in $\mathscr{R}$,

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \boldsymbol{J}^{-1}\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}. \tag{36}$$

4.1.1. Reconstruction of corner velocities

We consider the cell shown in Fig. 3. The four fluxes F_i, will be given on the edges E_i for $i = x0$, $x1$, $y0$, $y1$, and the normal vectors are defined in (18). Note that $\boldsymbol{n}_x(\hat{x},\hat{y}) = \boldsymbol{n}_x(\hat{x})$ and $\boldsymbol{n}_y(\hat{x},\hat{y}) = \boldsymbol{n}_y(\hat{y})$. Hence, we can define $\boldsymbol{n}_{x0} = \boldsymbol{n}_x(0)$, $\boldsymbol{n}_{x1} = \boldsymbol{n}_x(1)$, $\boldsymbol{n}_{y0} = \boldsymbol{n}_y(0)$, and $\boldsymbol{n}_{y1} = \boldsymbol{n}_y(1)$, respectively.

The corner velocities $\boldsymbol{q}_i$, $i = 1,\ldots,4$, will be solutions of 2×2 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ex(i)} = F_{Ex(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ey(i)} = F_{Ey(i)}, \end{cases} \quad i = 1,\ldots,4. \tag{37}$$

Here $Ex(i)$ and $Ey(i)$, $i = 1,\ldots,4$, refer to edges in the x- and y-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. This means that for $\boldsymbol{q}_1$, we get

$$\begin{cases} \boldsymbol{q}_1 \cdot \boldsymbol{n}_{x0} = F_{x0}, \\ \boldsymbol{q}_1 \cdot \boldsymbol{n}_{y0} = F_{y0}, \end{cases} \tag{38}$$

since E_{x0} and E_{y0} are adjacent to corner $\boldsymbol{x}_1$. The systems (37) are well-conditioned as long as the quadrilateral does not degenerate. If the fluxes have been computed exactly for a uniform flow field $\boldsymbol{q}$, then $\boldsymbol{q}_i = \boldsymbol{q}$.

Note that (37) implies that

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i) \cdot \boldsymbol{n}_i = F_i, \quad i = x0, x1, y0, y1, \tag{39}$$

where $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}(E_i)$ denotes $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ evaluated at a point on edge E_i. Hence, since the length of $\boldsymbol{n}_i$ is equal to the length of E_i, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ will reproduce the given edge fluxes.

4.2. Extension to 3D

We approximate $\boldsymbol{q}$ in (29) by a trilinear interpolation of the velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1,\ldots,8$,

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}} \equiv \sum_{i=1}^{8} \boldsymbol{q}_i \phi_i(\hat{x},\hat{y},\hat{z}). \tag{40}$$

4.2.1. Reconstructing corner velocities

Consider the cell in Fig. 4. The six fluxes F_i will be given on the faces S_i for $i = x0$, $x1$, $y0$, $y1$, $z0$, $z1$, respectively. The corresponding normal vectors $\boldsymbol{n}_i$ are defined in (31).

The flux integral of a velocity field $\boldsymbol{q}$ over the face S_i can be transformed to a double integral on a face of the unit cube in $\mathscr{R}$ using the trilinear transformation:

$$F_i = \int_0^1 \int_0^1 \boldsymbol{q} \cdot \boldsymbol{n}_i \,\mathrm{d}\alpha\,\mathrm{d}\beta, \quad i = x0, x1, y0, y1, z0, z1, \tag{41}$$

since the norm of $\boldsymbol{n}_i$ is equal to the surface Jacobian. If $\boldsymbol{q}$ is constant, we get,

$$F_i = \boldsymbol{q} \cdot \int_0^1 \int_0^1 \boldsymbol{n}_i(\alpha,\beta)\,\mathrm{d}\alpha\,\mathrm{d}\beta = \boldsymbol{q} \cdot \bar{\boldsymbol{n}}_i, \quad i = x0, x1, y0, y1, z0, z1, \tag{42}$$

where $\bar{\boldsymbol{n}}_i$, defined by the above equation, is given by a simple analytic expression, see e.g., [1]. Thus, in order to reproduce a uniform flow field, the corner velocities $\boldsymbol{q}_i$, $i = 1,\ldots,8$, will be solutions of 3×3 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sx(i)} = F_{Sx(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sy(i)} = F_{Sy(i)}, \\ \boldsymbol{q}_i \cdot \bar{\boldsymbol{n}}_{Sz(i)} = F_{Sz(i)}, \end{cases} \quad i = 1,\ldots,8. \tag{43}$$

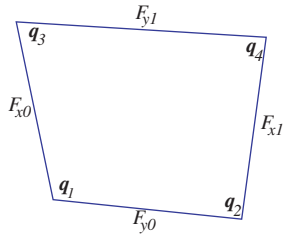


Fig. 3. Reconstructing velocities from fluxes in 2D.

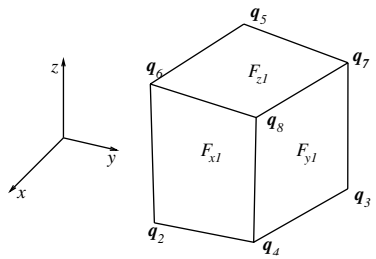


Fig. 4. Reconstructing velocities from fluxes in 3D.

Here $Sx(i)$, $Sy(i)$, and $Sz(i)$, $i = 1,\ldots,8$, refer to faces in the x-, y-, and z-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. In particular, for $\boldsymbol{q}_8$ we get

$$\begin{cases} \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{x1} = F_{x1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{y1} = F_{y1}, \\ \boldsymbol{q}_8 \cdot \bar{\boldsymbol{n}}_{z1} = F_{z1}. \end{cases} \tag{44}$$

Now, $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ and $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ are given by (40) and (36), respectively. As opposed to SFM and EFM, each component of the interpolated velocity field $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is a function of all three variables $\hat{x}, \hat{y}$, and $\hat{z}$. Therefore, analytical integration of $\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}$ is generally not possible.

Note that numerical integration of a velocity in $\mathscr{R}$ will not give the exact exit point from the cell, unless the last integration step ends on the cell boundary. Generally, interpolation is needed to determine the exit point [43].

4.3. Comparison of SFM, EFM, and CVI

Before summing up, let us compare the expressions of the EFM and CVI for the velocity field in 2D physical space. From Section 3.3 we know that

$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{y}) + F_{y1}\hat{y} \end{bmatrix}. \tag{45}$$

Furthermore, multiplication by $\boldsymbol{J}$ gives

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \boldsymbol{J}\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} k_1\hat{x}\hat{y} + k_2\hat{x} + k_3\hat{y} + k_4 \\ k_5\hat{x}\hat{y} + k_6\hat{x} + k_7\hat{y} + k_8 \end{bmatrix}, \tag{46}$$

for certain coefficients k_i, $i = 1,\ldots,8$, depending on the corners $\boldsymbol{x}_i$, $i = 1,\ldots,4$, and the edge fluxes F_i, $i = x0$, $x1$, $y0$, $y1$. Note that for incompressible flow (for which $F_{x0} - F_{y1} + F_{y0} - F_{x1} = 0$), both k_1 and k_5 vanish. Some manipulations show that

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \frac{1}{\det \boldsymbol{J}} \sum_{i=1}^{4} (\det \boldsymbol{J}(\boldsymbol{x}_i))\boldsymbol{q}_i \phi_i(\hat{x},\hat{y}), \tag{47}$$

where $\boldsymbol{q}_i$, $i = 1,\ldots,4$, are the corner velocities of the CVI method. Thus, the corner velocities for the CVI method and the EFM method are the same. It then follows that for parallelograms, where the determinant of the Jacobian $\boldsymbol{J}$ is constant, the CVI and EFM are identical. Also, since the Jacobian determinant is bilinear, it can be written as

$$\det \boldsymbol{J} = \sum_{i=1}^{4} (\det \boldsymbol{J}(\boldsymbol{x}_i))\phi_i(\hat{x},\hat{y}). \tag{48}$$

For a uniform flow field $\boldsymbol{q}$, we have, as noted after (38), that $\boldsymbol{q}_i = \boldsymbol{q}$. Therefore, it follows from (48) and (47) that $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{EFM}} = \boldsymbol{q}$. (Of course, by construction, the CVI method is also exact for uniform flow.) Equation (47) can only be extended to 3D for parallelepiped cells, since the corner velocities in 3D are determined from the average normals, $\bar{\boldsymbol{n}}_i$. The same is true for (48), since the Jacobian in 3D is not trilinear.

So far, we expect the main advantage of the CVI method to come with its extension to 3D. The CVI method is the only method that can reproduce uniform flow on irregular grids in 3D. In 2D, all the methods reproduce correct streamline paths for uniform flow, but only EFM and CVI reproduce the time-of-flight exactly. In the next section we demonstrate that CVI is the only method that converges for nonuniform grid refinements. We also show how the CVI method can be adapted to use half-edge fluxes. When the flow equations are solved by an MPFA method [1], half-edge fluxes are computed, and hence, more information about the velocity field is available. A disadvantage of CVI is that numerical integration must be used, whereas the EFM and SFM use analytical integration, which is faster.

5. Numerical experiments

In this section we compare the three methods, SFM, EFM, and CVI with respect to accuracy in producing both streamline paths and time-of-flight. Moreover, we will discuss their relative computer efficiency.

Errors in a streamline tracing method may be classified in several categories. First, there may be errors in the computed velocity field or in the computed fluxes that are used as input data. This will be the case in Section 5.4, where we present results on two quarter-five spot configurations; on the unit cube and for a simplified real field model. The other numerical experiments however, consider only velocity fields with analytic representations to eliminate errors due to approximate velocities. For such velocity fields, the streamlines can be calculated analytically or numerically to any desired accuracy. These exact streamlines will then be used to measure the errors in the different tracing methods.

Disregarding round-off errors in e.g., reconstruction of the corner velocities from fluxes, the remaining errors may come from the following sources

- Errors in the transformation of the velocity from $\mathscr{P}$ to $\mathscr{R}$.
- Errors in the interpolation in $\mathscr{R}$. These are related to:
  - assumptions on the normal component of the velocity at cell edges, or
  - the interpolation method (linear/bilinear/trilinear).
- Errors in the integration due to:
  - the evaluation of the Jacobian determinant, or
  - numerical integration of the velocity, or
  - interpolation to find the streamline exit points in the numerical integration.

These errors may combine or cancel each other at different parts of a given streamline. Due to the number of different error sources, and since the errors may cause the computed streamline to oscillate around, or diverge away from the true streamline, it is generally difficult to evaluate the accuracy and efficiency of a tracing method.

Error of the tracing method. Consider the exact streamline $\boldsymbol{s}(t;\boldsymbol{x}_j)$ and an approximate streamline $\boldsymbol{s}_h(t;\boldsymbol{x}_j)$ starting at the common point $\boldsymbol{x}_j$ at $t=0$, see Fig. 5. The length of the exact streamline at time t is given by

$$L(t) = \int_0^t \left|\frac{\mathrm{d}\boldsymbol{s}(t;\boldsymbol{x}_j)}{\mathrm{d}t}\right| \mathrm{d}t. \tag{49}$$

The streamlines are traced until they reach the outflow boundary of the physical domain. Let T be the time-of-flight coordinate of the streamline that first arrives at a boundary. We choose to evaluate the error in the approximate streamline by (see [32,43]):

$$\varepsilon_j(T) \equiv \frac{\|\boldsymbol{s}(T;\boldsymbol{x}_j) - \boldsymbol{s}_h(T;\boldsymbol{x}_j)\|}{L(T)}, \tag{50}$$

where $\|\cdot\|$ denotes Euclidean distance. This error measures the error per unit length in both streamline shape and time-of-flight.

The average error for a given method on a given grid for a number of starting points $\boldsymbol{x}_j$, $j=1,\ldots,N_s$, will be computed as

$$\bar{\varepsilon} = \frac{1}{N_s}\sum_{j=1}^{N_s} \varepsilon_j. \tag{51}$$

Random grids. In several of the test cases, irregular grids of hexahedral cells will be used. We start with a rectangular grid of $n_x \times n_y \times n_z$ uniformly partitioned grid cells. To obtain an irregular, randomly perturbed grid, each corner point in the uniform grid is perturbed by up to p percent relative to the grid cell size. Note that when these relative perturbations are retained for all refinements of the grid, these grids are referred to as rough grids. Refer to [31] for more details.

CVI integration method. To compute streamline paths, we have used the fourth-order, explicit Runge–Kutta solver in Matlab, ode45 [52]. This solver is based on the Dormand–Prince (4, 5) pair [16], which uses six function evaluations per time step. The equation solved for each grid cell is

$$\frac{\mathrm{d}\hat{\boldsymbol{s}}}{\mathrm{d}t} = \hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVI}}, \quad \hat{\boldsymbol{s}}(0) = \hat{\boldsymbol{s}}_0. \tag{52}$$

Since this is a method with step size control, we can choose a relative tolerance δ_r and an absolute tolerance δ_a. In each time step, the solver estimates the local error $\boldsymbol{e} = [e_1, e_2, e_3]$ in the solution $\hat{\boldsymbol{s}} = [y_1, y_2, y_3]$. This error must be less than or equal to the acceptable error, which is a function of the specified relative and absolute tolerances

$$|e_i| \leqslant \max(\delta_r|y_i|, \delta_a), \quad 1 \leqslant i \leqslant 3. \tag{53}$$

In the following we choose $\delta_a = 10^{-8}$ and use $10^{-8} \leqslant \delta_r \leqslant 10^{-3}$. If the time step is too large according to (53), the step is rejected, a smaller time step is computed, and a new set of function evaluations is needed. If the error is much smaller than the acceptable error, it is likely that the time step is unnecessarily small. The current step is therefore accepted and the next integration step is computed with a larger time step.

Generally, the last time step in a cell does not end on the cell boundary. Hence, interpolation is needed to determine the exit point [43]. The interpolation to find the exit point uses a fifth order Hermite polynomial [51,52] and a Newton iteration.

Finally, it is important to choose a good initial step size to avoid an excessive number of function evaluations [4]. For our purposes, we will not go into this topic, but assume that a good initial step has been found. This is done by picking an initial step that is too large for the given tolerance δ_r, and letting the solver iterate until an acceptable step is found.

It is also possible to integrate streamlines directly in physical space based on CVI. However, using a higher-order Runge–Kutta method like ode45 directly in physical space will generally be computationally expensive, since these methods require the velocity to be evaluated in the interior of each cell. At such points, the velocity is only given in reference coordinates and a velocity evaluation thus requires an inverse bilinear/trilinear transformation that is quite computationally expensive. In [23], we therefore suggest an alternative method based on a simple Euler predictor–corrector scheme that only requires velocities at cell boundaries in combination with a grid refinement scheme.

Reference streamlines. For comparing the tracing methods to an analytical solution, the exact streamline is computed using ode45 in Matlab with $\delta_r = 10^{-12}$ and $\delta_a = 10^{-12}$. The solution domain and the velocity is mapped to a reference element to simplify detection of domain boundaries.

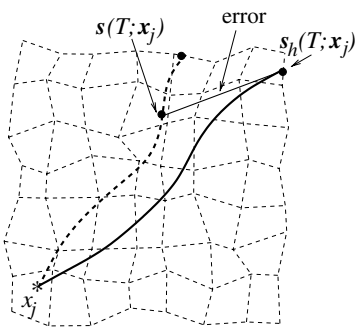


Fig. 5. Error of the method.

5.1. Uniform flow on random grids

We consider first four test cases for uniform flow on random grids using given constant velocity fields. Unless stated otherwise, the relative tolerance for the numerical integration in the CVI method is either $\delta_r = 10^{-3}$ or $\delta_r = 10^{-8}$, giving two methods denoted CVI3 and CVI8, respectively. The fluxes are computed analytically (to machine precision).

Case 1: *Perturbed grids*. For the first test we consider a uniform flow field $\boldsymbol{q} = [1,1,1]^{\mathrm{T}}$, and a base $10\times10\times10$ Cartesian grid partition of the unit cube. The simulation will

be performed on two series of perturbed grids, for which each vertex of the grid is moved a distance of up to p_k percent relative to the $h = 0.1$ grid spacing (i.e., on random grids with a p_k percent perturbation as defined in the previous subsection). For each grid and each tracing method, streamlines are traced from 100 random points in the front lower-left grid cell. The results of the test are shown in Fig. 6. For this, and all subsequent plots, we have plotted the (base 10) logarithm of the average error given in (51).

Note that for a constant velocity field the Runge–Kutta solver will only need a single time step for each cell independent of the value specified for δ_r. Thus, for velocity fields that are uniform or nearly uniform in a cell, we cannot assign any particular interpretation to the error tolerances used in the CVI method. However, since the velocity of the CVI method is integrated in reference space $\mathscr{R}$, this velocity will not be uniform in a cell even if the physical velocity field is uniform, unless the Jacobian matrix of the transformation is constant (i.e., the cells tend to parallelepipeds).

Fig. 6 shows that for small grid perturbations the errors of the CVI methods may be below their prescribed error tolerances, indicating that the velocity field in $\mathscr{R}$ is also uniform. As we reach a 1% perturbation, we begin to see the effect of the chosen relative tolerances for the CVI method. The results indicate that the error for the CVI method can be made arbitrarily small by decreasing δ_r and δ_a.

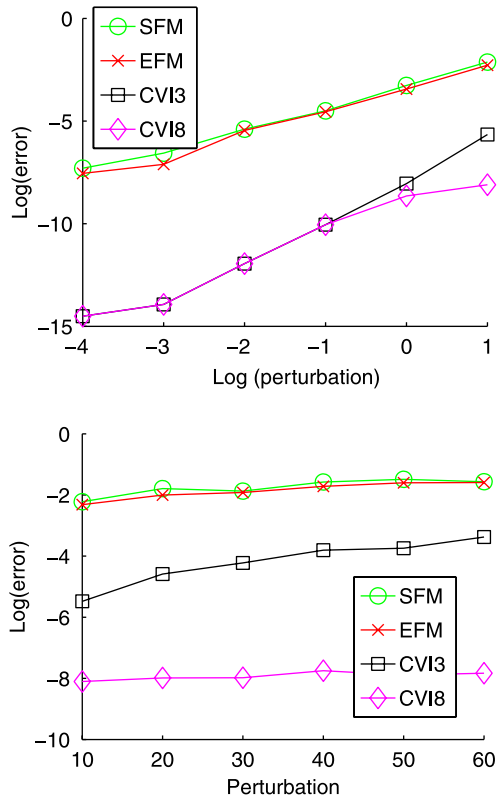


Fig. 6. (Case 1, uniform flow). Logarithm of error relative to the degree of irregularity of cells. Top: $\log_{10}(p_k) = -4, -3, \ldots, 1$. Bottom: $p_k = 10, 20, \ldots, 60$.

Case 2: *Uniform refinement.* For the second test we consider the error with respect to uniform grid refinement. We still consider a flow field $\boldsymbol{q} = [1,1,1]^{\mathrm{T}}$. Grid $\mathscr{G}_0$ consists of one grid cell and is a 50% random perturbation of the unit cube. Grid $\mathscr{G}_k$ for $k = 1,2,\ldots,6$ will be a trilinear map (see Section 3.4) of a uniform $2^k \times 2^k \times 2^k$ partition of the unit cube in $\mathscr{R}$ to the cell represented by $\mathscr{G}_0$, see Fig. 8 (left). The grids $\mathscr{G}_k$ will thus be refinements of $\mathscr{G}_0$ and asymptotically tend to a parallelepiped grid.

Streamlines are traced from 100 random points in the front lower-left grid cell of the finest grid ($\mathscr{G}_6$). The average error for grid $\mathscr{G}_k$, $k = 2,\ldots,6$, is shown in Fig. 7.

We observe that the CVI method is both more accurate and converges faster than EFM and SFM. For an $8 \times 8 \times 8$ refinement ($\mathscr{G}_3$), EFM and SFM have an error of approximately 10^{-3}. The same accuracy is achieved for the initial grid $\mathscr{G}_0$ (not shown in the plot) for the CVI3 method. For the given velocity field, the streamlines for EFM and SFM on grid $\mathscr{G}_3$, are traced through 21 cells. Consequently, SFM and EFM have to trace approximately 21 cells to achieve the same accuracy that CVI3 obtains on a single cell.

Moreover, we observe that the errors for the CVI methods are almost independent of the tolerance for $\mathscr{G}_k$, $k \geqslant 3$. The reason is that as the grid is refined and the grid cells tend to parallelepipeds, the velocity field in reference space also becomes approximately uniform, and the numerical integration will need only one time step for any δ_r.

Case 3: *Nonuniform refinement.* The setup will be the same as for Case 2, except that the refinement now is random. At each refinement level, the grids will be a 50% perturbation of a uniformly refined grid, see Fig. 8 (right). The average error for grid $\mathscr{G}_k$, $k = 2,\ldots,6$ is shown in Fig. 9. (The choice of a nonuniform refinement may seem strange, but is in fact what one will use for real field cases when trying to approach the resolution of an underlying geomodel, which will typically consist of highly irregular grid cells modeling complex geological structures.)

For this case, the velocity in reference space will not become uniform at the same rate as for the uniform refinement. Due to the nonlinearity of the velocity field in $\mathscr{R}$, the errors of the CVI method cannot be reduced further than

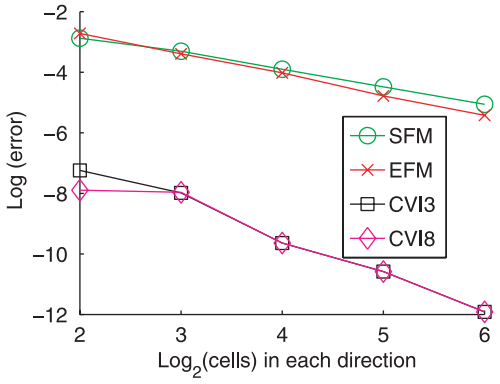


Fig. 7. (Case 2, uniform flow). Logarithm of error relative to uniform grid refinement.

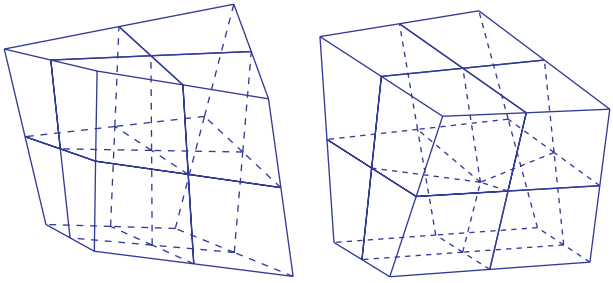


Fig. 8. Grids for the first refinement level ($\mathcal{G}_1$) in Case 2 (left) and Case 3 (right). Note that the base grid $\mathcal{G}_0$ is different for the two figures, and also only internal grid points are perturbed for Case 3, i.e., the central corner in the figure.

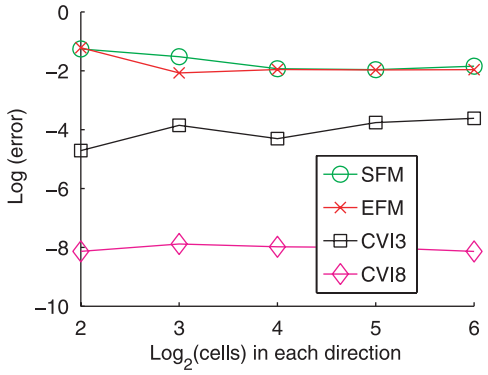


Fig. 9. (Case 3, uniform flow). Logarithm of error relative to nonuniform grid refinement.

the prescribed tolerances used in the numerical integration; i.e., 10^{-3} for CVI3 and 10^{-8} for CVI8, respectively. The errors are therefore almost independent of the grid spacing. To increase the accuracy of CVI, one must select a lower tolerance δ_r, which implies an increased number of integration steps. The situation is quite different for the SFM and EFM methods: Here the horizontal curves indicate that these methods do not converge as the grid is refined!

Case 4. In this test we consider error accumulation due to the length of the streamline and the number of cells traversed. For this case we use the flow field $\boldsymbol{q} = [1,0,0]^{\mathrm{T}}$. Grid $\mathcal{G}_k$ for $k = 0,1,\ldots,4$ will be a 50% perturbation of a $3 \cdot 2^k \times 3 \times 3$ partition of the parallelepiped $[0,2^k] \times [0,1] \times [0,1]$. Twenty streamlines are traced, each with a different random starting point in the central-left cell of each grid. Then the number of cells traversed for a given streamline for a given grid $\mathcal{G}_k$ will be approximately $3 \cdot 2^k$.

Fig. 10 shows the error per unit length for each streamline. Since this error is approximately constant for each method, we can in other words expect a uniform increase in the *absolute error* $\|\boldsymbol{s}(T;\boldsymbol{x}_j) - \boldsymbol{s}_h(T;\boldsymbol{x}_j)\|$ with increasing streamline length for all methods. Moreover, since the error constant is larger for EFM and SFM than for the CVI method, the accumulation of error with streamline length will be more pronounced for SFM and EFM.

We also see from (21) that the only difference between EFM and SFM is the scaling of the velocity vector. Thus the streamline shape will be the same for both methods,

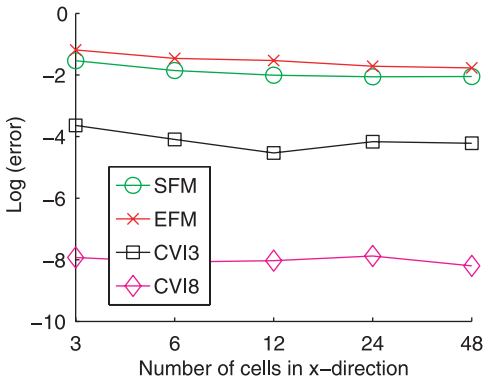


Fig. 10. (Case 4, uniform flow). Logarithm of error relative to length of streamline.

and the differences in the figure are only due to different time-of-flights. EFM uses a more accurate approximation of the Jacobian determinant than SFM and should therefore in principle be more accurate. Here, however, SFM is more accurate than EFM, which is explained by error cancellation in which the error in approximating the Jacobian determinant cancels the error in the interpolation.

5.2. Nonuniform flow on random grids

Streamline simulators use a sequential splitting method to solve the flow equations as explained in Section 2. For an incompressible flow in an isotropic and homogeneous medium with no sources, (2) reduces to the well-known Laplace equation. For this case it is easy to verify that

$$p(x,y,z) = \sin(\sqrt{2}x)\sinh(y)\sinh(z) \tag{54}$$

is a solution to (2). The corresponding velocity field is given by (4) as

$$\boldsymbol{q} = -\begin{bmatrix} \sqrt{2}\cos(\sqrt{2}x)\sinh(y)\sinh(z) \\ \sin(\sqrt{2}x)\cosh(y)\sinh(z) \\ \sin(\sqrt{2}x)\sinh(y)\cosh(z) \end{bmatrix}. \tag{55}$$

We will now consider the same type of tests as for uniform flow, using the velocity field in (55). The setup for the test cases will be similar to the cases for uniform flow in the previous section, except for the following:

- The analytical velocity field is now nonlinear, and numerical integration must be used to obtain fluxes. We have used a Lobatto quadrature [19] with a tolerance of 10^{-6}.
- The domain is shifted slightly to avoid the singularity of the velocity at the origin, see Fig. 11.
- Twenty streamlines are traced from the surface (see Fig. 11)

$$S = \{(x,y,z) : x = 0.5, 0.5 \leqslant y,z \leqslant 0.55\}.$$

Case 1: *Perturbed grids*. First we revisit the perturbed grids from Case 1 in Section 5.1, i.e., consider a p_k percent perturbation of the cells in a $10 \times 10 \times 10$ partition of the

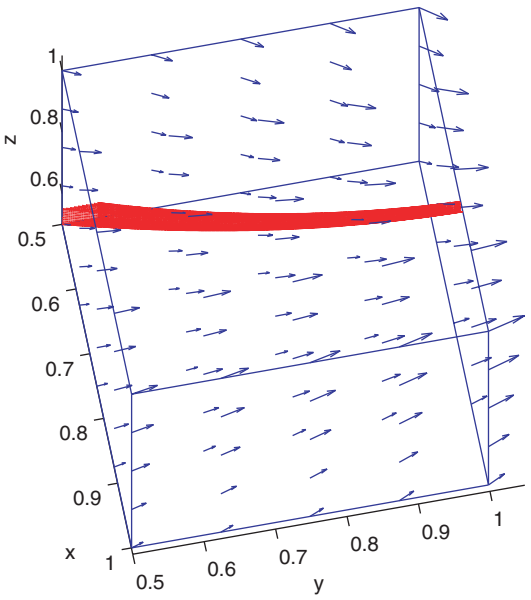


Fig. 11. Field plot for the velocity in (55). All the analytical streamlines traced for this case lie within the streamtube indicated in red.

cube shown in Fig. 11, but now with the velocity field (55). The results are shown in Fig. 12. The curves for CVI8 are identical to those of CVI3 and are therefore not shown in the plots. For small perturbations, the difference in the methods are overshadowed by the error in the interpolation since all methods use the same interpolation for small perturbations. As the perturbations increase, we see the differences in the methods. However, due to possible cancellation of errors (e.g., as discussed for Case 4 in Section 5.1), it is difficult to interpret these differences in favor of one method or the other, even though CVI generally has a slightly lower error than SFM and EFM.

Case 2: *Uniform refinement*. Next, we consider the error relative to a uniform refinement of a single skewed cell, as in Case 2 of Section 5.1. The results are shown in Fig. 13, where the CVI8 curve (not shown) is identical to the CVI3 curve. The figure shows that the convergence rates are equal for all three methods, indicating that the perturbations of the cells go faster to zero than the velocity approaches uniform flow in a given cell (otherwise, the CVI method would converge faster than SFM/EFM).

Case 3: Nonuniform refinement. Corresponding results for nonuniform refinement are shown in Fig. 14. This test case shows again the benefit of the CVI method when the flow becomes approximately uniform in each cell and the grid cells are kept irregular: As for Case 3 in Section 5.1 for uniform flow, SFM and EFM do not seem to converge as the grid is refined.

Note, however, that the curves depend on a particular realization of the random grids. Ideally, we should have averaged over a set of such realizations. This may explain

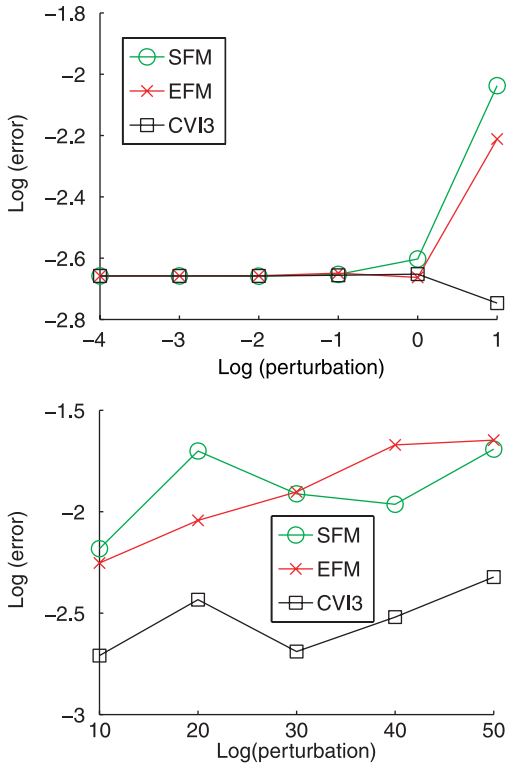


Fig. 12. (Case 1, nonuniform flow). Logarithm of the error relative to the degree of irregularity of cells Top: $\log_{10}(p_k) = -4, -3, \ldots, 1$. Bottom: $p_k = 10, 20, \ldots, 50$.

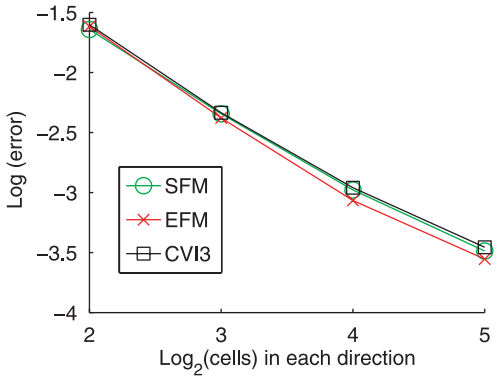


Fig. 13. (Case 2, nonuniform flow). Logarithm of error relative to uniform grid refinement.

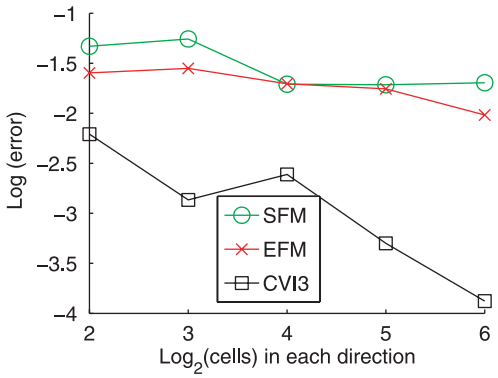


Fig. 14. (Case 3, nonuniform flow). Logarithm of error relative to nonuniform grid refinement.

the irregular behavior of the CVI curve. Also, since the fluxes are computed numerically with a tolerance of 10^{-6}, we cannot expect the same level of accuracy as the grid is refined for the CVI methods compared to Case 3 for uniform flow, where the fluxes were computed analytically.

5.3. Analytic flow on truncated pyramidal grids

We now study the effect of using a special kind of grid, denoted a *truncated pyramidal grid*, see also [40]. The construction of such a grid is illustrated by Fig. 15, where the entire grid conforms to a cubic domain. As seen here, the inner cells are truncated pyramids, and 'infill' cells are used to account for the boundary, and the cells will be turned upside down systematically from layer to layer. The surface with the smallest area of a truncated pyramid will either be the roof surface or floor surface of the cells, and the ratio between the smallest and the largest surface area is kept constant and equal to 1/9 as the grids are refined.

As pointed out in the description of the CVI method (Section 4), this method is constructed to reproduce uniform flow exactly regardless of the grid geometry. SFM and EFM, on the other hand, do not reproduce uniform flow and will therefore, as observed from the results below, produce a systematic error for skewed cells like the truncated pyramids.

Case 1: *Uniform flow*. Fig. 16 shows the performance of the three tracing methods on a sequence of refinements of truncated pyramidal grids for the uniform flow field $\boldsymbol{q} = [0,0,1]$ on the cubic domain $\{1 \leqslant x,y,z \leqslant 2\}$. For each refinement, 100 streamlines are traced from randomly distributed points on the bottom surface ($z = 1$) of the domain until they exit at the top of the domain ($z = 2$). We see that the errors of the CVI method stay at the level of their prescribed tolerances, indicating that the velocity in the reference space is not becoming uniform as the grid is refined

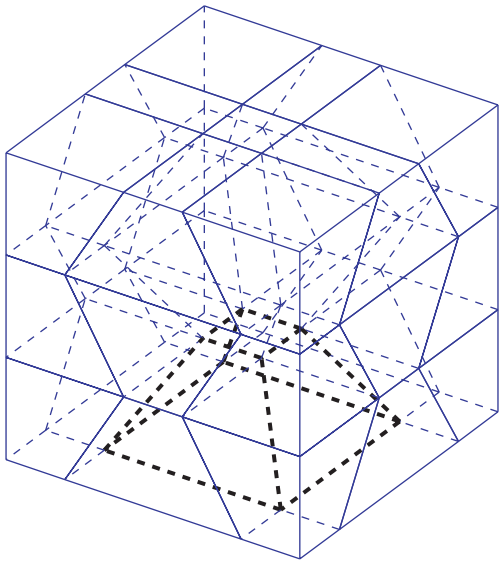


Fig. 15. Sketch of a truncated pyramidal grid.

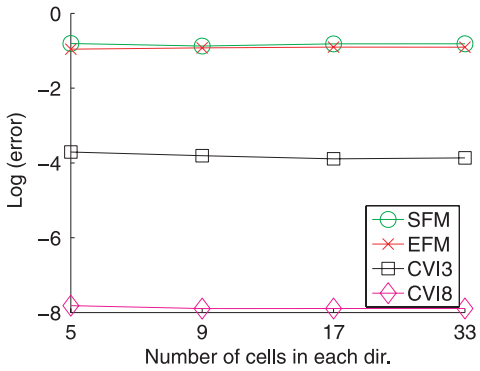


Fig. 16. (Case 1, truncated pyramids). Errors versus grid refinement.

(compare with Fig. 7). We also notice that the streamlines of SFM and EFM do not converge as the grid is refined.

Case 2: *Combined uniform and nonuniform flow*. We next replace the uniform flow field by a combination of nonuniform and uniform flow, obtained by taking the negative gradient of the pressure field

$$p(x,y,z) = \cosh(x)\cos(y) + cz,$$

which is an analytical solution to the Laplace equation that here represents a simplification of the general pressure equation (2). Note that this pressure field gives rise to a uniform flow in the z-direction since the z-component of the gradient of the pressure field is a constant. This situation can occur if gravity is the driving force for the flow in the reservoir and the permeability of the medium is almost homogeneous. Then the flow locally is almost uniform in the z-direction of the reservoir.

Fig. 17 depicts how the methods behave for refinements of the truncated pyramidal grids introduced above, where the constant $c = 5$ is used and 20 streamlines are traced for each refinement level. The CVI method converges with the expected rate, whereas the convergence for SFM and EFM is both slower and decays significantly for the highest refinements, suggesting that these two methods may fail to converge asymptotically.

Case 3: *Nonuniform flow*. Finally, we study the effect of a gradual change from uniform flow in the z-direction to

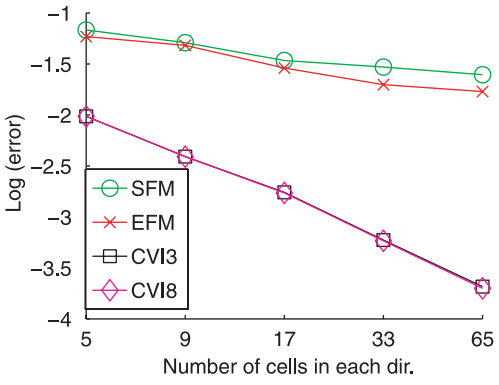


Fig. 17. (Case 2, truncated pyramids). Errors of streamlines versus grid refinement.

nonuniform flow in all directions. This is done by allowing the σ-parameter to increase for the solution

$$p(x,y,z) = \cosh(x)\cos(y) + cz + \sigma\cosh(x)\cos(z). \tag{56}$$

The result for an $11\times11\times11$ grid is presented in Fig. 18, where we have plotted the $\log_{10}\sigma$ along the x-axes. As expected, the errors decrease as the flow approaches uniform in the z-direction (i.e., $\sigma\to0$). Repeating the refinement process from Case 2, we observed that the CVI method still converges for a fixed σ and increasing refinements, but the error constant increases for increasing values of σ.

Summing up, when either truncated pyramids or general rough grids are used in 3D, the difference between our new streamline method and SFM and EFM is striking. The CVI method is superior to the other methods in terms of accuracy; this is manifested as either loss of convergence or diminished convergence rates for the SFM and EFM methods.

5.4. Quarter-five spot

In this subsection we use a control-volume finite-difference method [1] to solve the single-phase flow equations,

$$\begin{aligned} &\operatorname{div}\boldsymbol{q} = -\operatorname{div}(\boldsymbol{K}\operatorname{grad}p) = f \quad &&\text{in } \Omega,\\ &\boldsymbol{q}\cdot\boldsymbol{n} = 0 &&\text{on } \partial\Omega, \end{aligned} \tag{57}$$

and provide discrete fluxes on cell interfaces. Here, $\boldsymbol{q} = -\boldsymbol{K}\operatorname{grad}p$ is the Darcy velocity, p is the pressure, $\boldsymbol{K}$ is the permeability tensor, f is a source term, Ω is the solution domain, $\partial\Omega$ is its boundary, and $\boldsymbol{n}$ is the unit outward normal to $\partial\Omega$. The domain will be discretised with different kinds of $n_x\times n_y\times n_z$ grids.

A 3D extension of the classical 2D quarter-five spot test case is generated by placing an injector and a producer, respectively, in the lower and upper corner cells $(1,1,1)$ and (n_x,n_y,n_z). Streamlines will then be traced from the injector to the producer. As an error measure, we will in this subsection only compare the time-of-flight in the producing wells. To this end, we use a relative error measure by dividing the absolute difference in time-of-flights by the time-of-flight of the reference streamline. These relative errors are then averaged over all the traced streamlines.

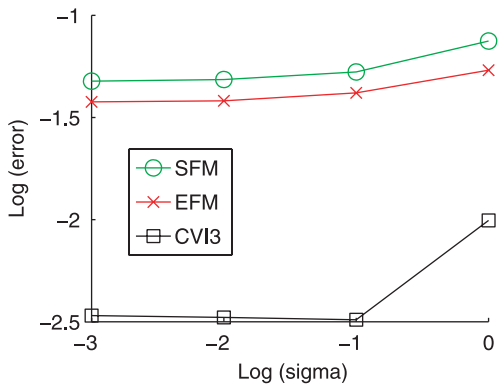


Fig. 18. (Case 3, truncated pyramids). Effect of flow becoming gradually more nonuniform by increasing σ in Eq. (56) when the grid resolution is fixed.

Case 1: *Random grids*. Let Ω be the unit cube and assume that $\boldsymbol{K}\equiv1$ over the whole domain. On this simplified medium, we compare the streamline methods on a coarse $10\times10\times10$ random grid. The inner corners of the coarse grid are perturbed such that the grid is nonorthogonal throughout the simulation domain. Wells are implemented by using nonhomogeneous Neumann conditions at the faces of the corner cells $(1,1,1)$ and $(10,10,10)$. We trace streamlines starting from 225 uniformly spaced entry points along the inflow boundary given by the boundary faces of cell $(1,1,1)$. The streamlines are traced until they exit at the outflow boundary, i.e., at some point of the boundary faces of cell $(10,10,10)$.

Time-of-flights at the producer are compared to the reference case simulated by the SFM method on a $50\times50\times50$ uniform Cartesian grid, where the entry points agree with the entry points of the coarse grid. Table 1 reports the average errors and corresponding standard deviations for the three tracing methods. As seen from the table, the CVI method performs slightly better than the other two methods; the average error is roughly 20% smaller for the CVI method compared to e.g., SFM.

Case 2: *A simplified field case*. Next, we study the streamline generation on a simplified version of a real field model. We use a 3D grid that is an extension of a 2D grid where the height of the top and bottom of the medium varies throughout. The grid has moderate grid aspect ratios that resemble typical features of field cases; here this ratio is approximately 1/20 for height versus length of a typical grid block. A reference solution will be generated on a fine $45\times45\times45$ grid, which is a uniform refinement of a coarse $9\times9\times9$ grid on which the tracing methods will be compared. The two grids are compatible in the sense that each cell interface of the coarse grid is exactly matched by a set of interfaces in the fine grid. To compute a reference solution we solve the flow equations on the fine grid with no-flow boundary conditions and a nonzero source f to simulate wells. The approximate solution on the coarse grid are found by simple averaging of fine-grid fluxes.

We will apply two different permeability fields; first a homogeneous permeability field, and then a layered log-normal permeability field that has similarity to real field cases. For both cases, three hundred streamlines are traced from the injector to the producer. The reference solution will be traced with SFM on the $45\times45\times45$ grid.

For the homogeneous case, $\boldsymbol{K}\equiv1$, averaged errors and associated standard deviations for the three methods are presented in Table 2. We see that there is an apparent improvement for CVI compared to SFM and EFM, and both errors and standard deviations are significantly smaller.

For the heterogeneous case, the permeability field is defined on the coarse $9\times9\times9$ grid. On the fine grid, the permeability is therefore constant on patches of $5\times5\times5$ cells. We apply an isotropic, layered log-normally distrib-

Table 1
Average errors and standard deviations for the different tracing methods. Random grid perturbations of 16%

Method	Average error	Std. dev.
SFM	0.022	0.015
EFM	0.021	0.015
CVI3	0.017	0.012

Table 2
Average errors and associated standard deviations for the three tracing methods for the homogeneous case

Method	Average error	Std. dev.
SFM	0.084	0.076
EFM	0.083	0.077
CVI3	0.077	0.068

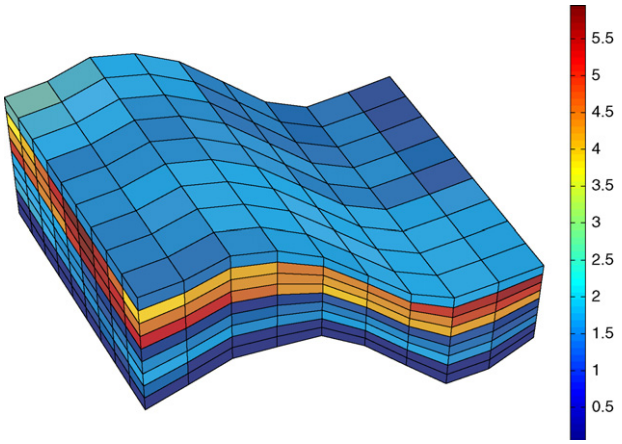


Fig. 19. Logarithm of the horizontal permeability for the simplified 3D field model.

uted permeability field, see Fig. 19. Fig. 20 depicts 75 of the streamlines traced on the fine grid, and Fig. 21 shows the base 10 logarithm of the time-of-flight for the reference solution and for the approximate solutions.

Note that in Fig. 21 the red[1] SFM curve and the black EFM curve are more or less covered by the magenta CVI3 curve. Due to the large variation in permeability on the coarse $9\times9\times9$ grid, we observe huge variations in time-of-flight for different streamlines. Unfortunately, there are equally large differences if the same streamline is traced using coarse and fine-grid fluxes, respectively, which indicates nonlinearity in the velocity field and a high information loss in the flux averaging. This shows the futility of comparing different streamline methods on 'realistic' models, that is for skew grids and permeability fields with complex heterogeneity structures. In this paper, we have therefore mainly focused on simplified models, on which one has control of the reference solution and errors in

[1] For interpretation of color in figures, the reader is referred to the Web version of this article.

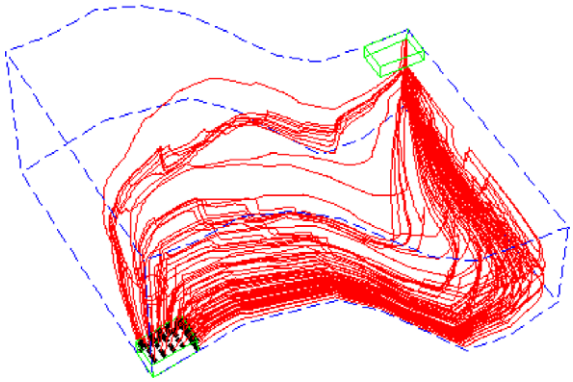


Fig. 20. Seventy-five streamlines traced using the fine-grid velocity for the layered permeability field shown in Fig. 19. Injector and producer cells for the coarse grid are shown in green. Black dots indicate starting points for the streamlines.

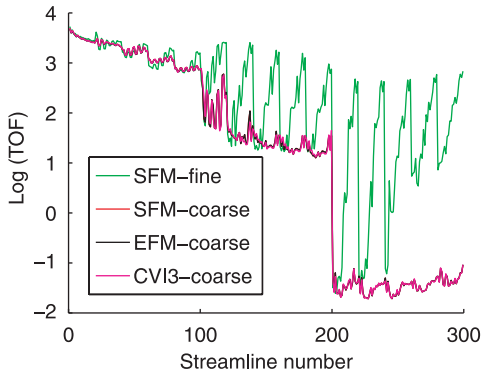


Fig. 21. Time-of-flight distribution for SFM on the fine grid, and for SFM, EFM, and CVI3 on the coarse grid. The EFM and CVI3 results on the fine grid are not shown since they could not be distinguished from the SFM curve on the fine grid. Note that the axis have been scaled: the ratio between height and length of a typical grid block is really 1/20. The streamlines are distributed uniformly over the faces rather than being distributed uniformly according to their associated amount of fluid flux.

the velocity fields on the coarser grids. More details can be found in [41], where a discontinuous Galerkin method was used for computing time-of-flight.

5.5. Runtime comparisons

Finally, we investigate the runtimes of the different methods. For the CVI method, there will be a relation between the number of time steps needed per cell and the prescribed tolerance δ_r. Tables 3 and 4 report the average number of time steps per cell for different values of δ_r and different perturbations of the grids for Case 1 from Sections 5.1 and 5.2.

Table 3
Relation between number of time steps per cell and δ_r for a 60% perturbation for Case 1 for uniform and nonuniform flow

Tolerance δ_r	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Uniform	1.3	1.7	2.3	3.0	4.2	5.2
Nonuniform	1.4	1.9	2.5	3.2	4.4	5.4

Table 4
Relation between cell perturbation and number of time steps per cell for a given δ_r for Case 1 for uniform and nonuniform flow

Perturbation in %	0	10	20	30	40	50	60
Uniform ($\delta_r = 10^{-3}$)	1.0	1.0	1.0	1.1	1.1	1.2	1.3
Uniform ($\delta_r = 10^{-8}$)	1.0	2.3	3.2	3.6	4.2	4.6	5.2
Nonuniform ($\delta_r = 10^{-3}$)	1.0	1.0	1.0	1.0	1.1	1.2	1.4
Nonuniform ($\delta_r = 10^{-8}$)	1.1	2.2	3.0	3.4	3.8	4.4	5.4

5.5.1. Timing results

Let us now compare the computational costs required to determine the exit point given a starting point within the cell. If we normalize the results such that SFM uses one time unit on average, EFM uses 1.5 time units, and CVI uses eight time units (assuming one time step per cell).

The runtime of CVI is essentially made up of the time used for:

- function evaluations,
- interpolation to find the exit point, and
- the reconstruction of the corner velocities.

By a function evaluation we mean evaluation of the velocity at a given point in reference space. This is done, as shown in Section 4, by a trilinear interpolation to find the velocity in physical space. Then the velocity in reference space is found by a multiplication by the inverse of the Jacobian matrix. The evaluation of the inverse is quite time-consuming, so a function evaluation takes approximately one time unit. One integration step generally requires six function evaluations, but for the first integration step in a cell, we need one extra function evaluation to get started. The interpolation to find the exit point uses a fifth-order polynomial [52] and a Newton iteration. This part consumes approximately one half time unit. Finally, reconstruction of corner velocities from discrete fluxes also takes one half time unit. Altogether, this consumes eight time units.

Although the CVI method is slower, the accuracy is higher when the flow becomes uniform. Our test cases for uniform flow indicate that to reach the same level of accuracy, SFM and EFM must use a refined grid, for which the computational cost is higher, in particular if refining the grid requires a new global pressure solution since the cost of streamline tracing typically is minuscule compared with the cost of solving the pressure equation. Moreover, in [23] we showed that by integrating directly in physical space rather than in reference space, the CVI method can be made as efficient as EFM for cases with uniform and almost uniform flow in 2D.

5.6. Using CVI with half-edge fluxes

The CVI method interpolates velocities in a manner that makes it appealing to use a *finer resolution* of the velocities. Moreover, if fluxes/velocities are evaluated for the half-edge rather than for the full edge, this subresolution in the fluxes may possibly be taken advantage of.

We have tested the CVI method with half-edge fluxes (denoted CVIH) in 2D for the velocity field,

$$\boldsymbol{q} = \begin{bmatrix} \sinh x \cos y \\ -\cosh x \sin y \end{bmatrix}, \tag{58}$$

which is an analytical solution to the Laplace equation. When using CVI with half-edge fluxes, the interpolated velocity field is still given by (35), but the corner velocities will be different. Equation (37) is replaced by

$$\begin{cases} \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ex(i)} = 2F_{Ex(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{Ey(i)} = 2F_{Ey(i)}, \end{cases} \quad i = 1, \ldots, 4, \tag{59}$$

where $F_{Ex(i)}$ and $F_{Ey(i)}$ are now *half*-edge fluxes adjacent to corner i. Eq. (59) implies that the normal component of $\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVI}}$ varies *linearly* along each edge.

Case 1: *Half-edge fluxes on parallelogram grids*. For parallelogram grids, CVI, SFM, and EFM are identical, as noted in Section 4.3. When the skewness of the parallelograms gets large, these methods will produce streamlines with apparent cusps. The CVIH method will reduce these artifacts, see Fig. 22.

Case 2: *Comparing BDM and CVIH*. A mixed finite-element method [7] can be employed to solve the pressure equation for both pressure and velocity, in which case the velocity must belong to the space $H(\mathrm{div};\Omega)$. A divergence-free higher-order velocity field can then be used to approximate $H(\mathrm{div};\Omega)$. For two degrees-of-freedom per edge in the two-dimensional case, the velocity should belong to the Brezzi–Douglas–Marini (BDM) space of order one, with the added condition of zero divergence [36,37]. By the BDM method we will in the following mean streamlines traced by the above velocity field. See [38,39,27] for a comprehensive discussion of streamline tracing in 2D using low and high-order mixed finite-element velocity spaces.

In our final example, we compare all the methods introduced so far: BDM, CVIH, CVI, SFM, and EFM for the analytical solution in (58). We start with a 10×10 uniform

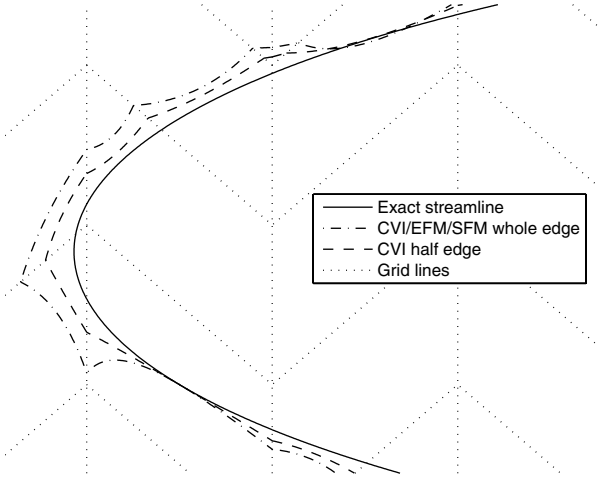


Fig. 22. Using CVI with half-edge fluxes. Note that the aspect ratio of the plot is 1:2.75, i.e., the y-direction is compressed. A horizontal line really makes an angle of 71° with the bottom or top edge of the parallelograms.

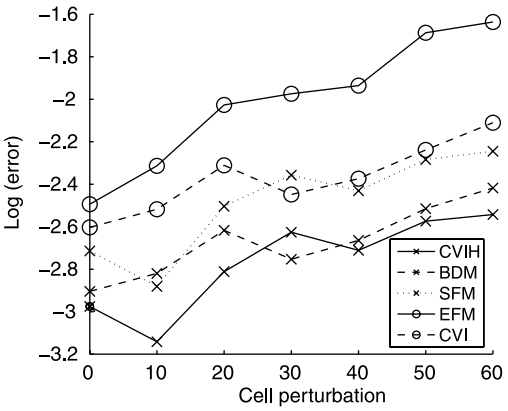


Fig. 23. Comparing different methods. Base 10 logarithm of error.

partition of $[0,1]\times[0,1]$ and consider grids obtained by a $10i$ percent perturbation for $i = 1,\ldots,6$. Exact edge and half-edge fluxes will be computed for each grid. These fluxes will be used to construct interpolated velocity fields for all the methods. For each grid, streamlines are started from 20 random points within a random cell. This cell is selected randomly as one of three cells in the top row of the grid. The results are shown in Fig. 23. Here, as also remarked for Case 4 in Section 5.1, SFM is better than EFM due to cancellation of errors. We also note that the CVIH and BDM methods are of approximately the same accuracy.

6. Summary and concluding remarks

This work has investigated streamline generation on irregular grids in 3D and has in particular focused on the problem of representing uniform flow on hexahedral grids. We have considered two standard methods, the standard flux-mapping method (SFM) [13,44,46] and the extended flux-mapping method (EFM) [25]. For irregular grids, these methods are based on a trilinear transformation of each grid cell to a unit cube together with a linear interpolation scaled by the Jacobian. The two methods only differ in the way they treat the Jacobian; see Eqs. (20) and (21). The major advantage of the SFM and EFM methods is that they both allow for fast and analytical integration of streamlines due to the linear flux interpolation. On the other hand, the methods share the same fundamental deficiency of being inaccurate for irregular (and rough) grids. That is, for irregular grids, the interpolated velocity field used in these methods will generally depend on the geometry of the grid cell, except for the 2D case, where a linear flux interpolation in reference space is sufficient to model the normal vectors $\boldsymbol{n}_x$ and $\boldsymbol{n}_y$ exactly. In 3D, the flux of a uniform flow field varies quadratically for a general hexahedral cell, and also a flux interpolation cannot capture the variation of the normal vectors that describe the geometry of the cell.

To improve the tracing accuracy, we introduced a new method, which we called the corner-velocity interpolation (CVI) method. Instead of interpolating the velocity inside the cell based on discrete fluxes at cell edges, the method interpolates the velocity based on (reconstructed) velocities at the eight corner points of a hexahedral cell. This way, we get a method that is generally less sensitive to the regularity of the cells and in particular is able to reproduce uniform flow regardless of the cell geometries, a property that is considered to be of high importance in solution of (elliptic) pressure equations. Moreover, the CVI method can easily be adapted to exploit the extra accuracy represented in half-edge fluxes, when these are available (see Section 5.6).

In Section 5, we compare the three tracing methods on a variety of simple test cases, focusing in particular on representing uniform flow in 3D. Our test cases in Sections 5.1,5.2 and 5.3 can be divided in two categories. In the first category, the irregularity of the grids diminish as they are refined. For these grids, all three methods converge, but CVI typically converges faster and is more accurate on each specific grid as the flow becomes uniform.

In the second category, we consider so-called *rough grids* and nonuniform refinements of these. (Nonuniform refinement may seem a bit strange on a first glance, but is in fact what will typically be used when refining a coarse simulation model towards an underlying geological model, which typically contains highly irregular cells used to model the complex structures of the underlying geology.) The test cases in this category reveal significant differences in the behavior of the three methods. For tests with uniform flow, the SFM and EFM methods do not only fail to reproduce uniform flow; they also fail to converge as the grids are (nonuniformly) refined. For nonuniform flow cases, our tests establish convergence of the CVI method, whereas the SFM and EFM methods either do not converge asymptotically or have a small convergence rate for the refinements levels observed.

The added accuracy of the CVI method comes at the cost of an increased computational complexity, which is imposed by the need to use numerical integration of a set of ODEs to compute streamlines. Since EFM and SFM use analytical integration, these methods will be significantly faster. We are therefore investigating various means to speed up the CVI method, for instance, integrating streamlines directly in physical space using an adaptive Euler predictor–corrector method [23]. The method proved to be quite successful in 2D, but has not yet been extended to 3D. Another tempting idea is to use a hybrid method, in which SFM is used for regular cells and CVI is used for irregular cells. This, however, requires some kind of error estimate for SFM and EFM and remains to be investigated.

Acknowledgements

The authors would like to thank Jan Nordbotten and two anonymous referees for useful comments and suggestions for improving the manuscript.

The research of K.–A. Lie and H. Hægland was funded by the Research Council of Norway under grants no. 158908/I30 and 173875/I30, respectively.

References

[1] Aavatsmark I. An introduction to multipoint flux approximations for quadrilateral grids. Comput Geosci 2002;6:405–32.

[2] Batycky RP. A three-dimensional two-phase field scale streamline simulator. PhD thesis, Stanford University, Department of Petroleum Engineering, 1997.

[3] Bear J. Dynamics of fluids in porous media. Environmental science series. New York: American Elsevier Publishing Company; 1972.

[4] Bensabat J, Zhou Q, Bear J. An adaptive pathline-based particle-tracking algorithm for the Eulerian–Lagrangian method. Adv Water Resour 2000;23:383–97.

[5] Bratvedt F, Bratvedt K, Buchholz CF, Gimse T, Holden H, Holden L, et al. Frontline and frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking. Surveys Math Indust 1993;3:185–215.

[6] Bratvedt F, Gimse T, Tegnander C. Streamline computations for porous media flow including gravity. Transp Porous Media 1996;25(1):63–78.

[7] Brezzi F, Fortin M. Mixed and hybrid finite element methods. Springer series in computational mathematics, vol. 15. New York: Springer-Verlag; 1991.

[8] Cai Z, Jones JE, McCormick SF, Russell TF. Control-volume mixed finite element methods. Comput Geosci 1997;1:289–315.

[9] Charbeneau RJ, Street RL. Modeling groundwater flow fields containing point singularities: Streamlines, travel times, and breakthrough curves. Water Resour Res 1979;15(6):1445–50.

[10] Cheng H-P, Cheng J-R, Yeh G-T. A particle tracking technique for the Lagrangian–Eulerian finite element method in multi-dimensions. Int J Numer Methods Engrg 1996;39:1115–36.

[11] Cheng H, Osako I, Datta-Gupta A, King MJ. A rigorous compressible streamline formulation for two- and three-phase black-oil simulation. Paper SPE 96866 in Proceedings of the SPE annual technical conference and exhibition, Dallas, Texas, 2005.

[12] Ciarlet PG. The finite element method for elliptic problems. Amsterdam: North-Holland; 1980.

[13] Cordes C, Kinzelbach W. Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements. Water Resour Res 1992;28(11):2903–11.

[14] Crane M, Bratvedt F, Bratvedt K, Childs P, Olufsen R. A fully compositional streamline simulator. SPE 63156. In: Proceedings of the SPE annual technical conference and exhibition, Dallas, Texas, 1–4 October, vol. 3, 2000.

[15] Di Donato G, Blunt MJ. Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs. Water Resour Res 2004;40(4).

[16] Dormand JR, Prince PJ. A family of embedded Runge–Kutta formulae. J Comput Appl Math 1980;6(1):19–26.

[17] Durlofsky LJ. Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities. Water Resour Res 1994;30(4):965–73.

[18] Edwards MG. Unstructured, control-volume distributed full-tensor finite-volume schemes with flow based grids. Comput Geosci 2002;6:433–52.

[19] Gander W, Gautschi W. Adaptive quadrature – revisited. BIT 2000;40:84–101.

[20] Gmelig Meyling RHJ. A characteristic finite element method for solving non-linear convection-diffusion equations on locally refined grids. In: Proceedings of the second European conference on the mathematics of oil recovery, 11–14 September, Arles, France, 1990.

[21] Gmelig Meyling RHJ. Numerical methods for solving the nonlinear hyperbolic equations of porous media flow. In: Proceedings of the third international conference on hyperbolic problems, Uppsala, Sweeden, 1990.

[22] Goode DJ. Particle velocity interpolation in block-centered finite difference groundwater flow models. Water Resour Res 1990;26(5):925–40.

[23] Hægland H, Dahle HK, Aavatsmark I, Eigestad GT, Lie K-A. Adaptive streamline tracing for streamline simulation on irregular grids. In: Proceedings of the XVI international conference on computational methods in water resources, June, Denmark, 2006.

[24] Ingebrigtsen L, Bratvedt F, Berge J. A streamline based approach to solution of three-phase flow. SPE 51904. In: SPE reservoir simulation symposium, Houston, Texas, 14–17 February 1999.

[25] Jimenez E, Sabir K, Datta-Gupta A, King MJ. Spatial error and convergence in streamline simulation. SPE 92873. In: SPE reservoir simulation symposium, Woodlands, Texas, 31 January–2 February 2005.

[26] Juanes R, Lie K-A. A front-tracking method for efficient simulation of miscible gas injection processes. SPE 93298. In: SPE reservoir simulation symposium, Woodlands, Texas, 31 January–2 February 2005.

[27] Juanes R, Matringe SF. Unified formulation of velocity fields for streamline tracing on two-dimensional unstructured grids. Comput Methods Appl Mech Engrg, submitted for publication.

[28] Kaasschieter EF. Mixed finite elements for accurate particle tracking in saturated groundwater flow. Adv Water Resour 1995;18(5):277–94.

[29] King MJ, Datta-Gupta A. Streamline simulation: a current perspective. In Situ 1998;22(1):91–140.

[30] Kipfer P, Reck F, Greiner G. Local exact particle tracing on unstructured grids. Comput Graph Forum 2003;22(2):1–9.

[31] Klausen RA, Winther R. Robust convergence of multi point flux approximation on rough grids. Numer Math 2006;104(3):317–37.

[32] Knight D, Mallinson G. Visualizing unstructured flow data using dual stream functions. IEEE Trans Vis Comput Graph 1996;2(4):355–63.

[33] Lebedev LP, Cloud MJ. Tensor analysis. River Edge, NJ: World Scientific; 2003.

[34] Lie K-A, Juanes R. A front-tracking method for the simulation of three-phase flow in porous media. Comput Geosci 2005;9(1):29–59.

[35] Lu N. A semianalytical method of path line computation for transient finite-difference groundwater flow models. Water Resour Res 1994;30(8):2449–59.

[36] Matringe SF. Accurate streamline tracing and coverage. Master's thesis, Stanford university, Dept of petr eng, 2004.

[37] Matringe SF, Gerritsen MG. On accurate tracing of streamlines. In: Proceedings of the SPE annual technical conference and exhibition, Houston, Texas, 26–29 September 2004.

[38] Matringe SF, Juanes R, Tchelepi HA. Streamline tracing on general triangular and quadrilateral grids. Paper SPE 96411. In: Proceedings of the SPE annual technical conference and exhibition, Dallas, Texas, 2005.

[39] Matringe SF, Juanes R, Tchelepi HA. Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids. J Comput Phys, in press, doi:10.1016/j.jcp.2006.07.004.

[40] Naff RL, Russell TF, Wilson JD. Shape functions for velocity interpolation in general hexahedral cells. Comput Geosci 2002;6(3–4):285–314.

[41] Natvig JR, Lie K-A, Eikemo B, Berre I. A discontinuous Galerkin method for computing single-phase flow in porous media, submitted for publication.

[42] Oliveira A, Baptista AM. On the role of tracking on Eulerian–Lagrangian solutions of the transport equation. Adv Water Resour 1998;21:539–54.

[43] Pokrajec D, Lazic R. An efficient algorithm for high accuracy particle tracking in finite elements. Adv Water Resour 2002;25:353–69.

[44] Pollock DW. Semi-analytical computation of path lines for finite-difference models. Ground Water 1988;26(6):743–50.

[45] Prévost M. Accurate coarse reservoir modeling using unstructured grids, flow-based upscaling and streamline simulation. PhD thesis, University of Stanford, 2003. http://geothermal.stanford.edu/pereports/search.htm.
[46] Prévost M, Edwards MG, Blunt MJ. Streamline tracing on curvilinear structured and unstructured grids. SPE J 2002(June):139–48.
[47] Russell TF, Healy RW. Analytical tracking along streamlines in temporally linear Raviart-Thomas velocity fields. In: Proceedings of the XIII international conference on computational methods in water resources, Alberta, Canada, 25–29 June 2000.
[48] Sadarjoen IA, van Walsum T, Hin AJS, Post FH. Particle tracing algorithms for 3d curvilinear grids. In: Nielson GM, Hagen H, Müller H, editors. Scientific visualization: overviews, methodologies, and techniques. Los Alamitos, Calif: IEEE Computer Society; 1997. p. 311–35 [chapter 14].
[49] Schafer-Perini AL, Wilson JL. Efficient and accurate front tracking for two-dimensional groundwater flow models. Water Resour Res 1991;27(7):1471–85.
[50] Shafer JM. Reverse pathline calculation of time-related capture zones in nonuniform flow. Ground Water 1987;25(3):283–9.
[51] Shampine LF. Interpolation for Runge–Kutta methods. SIAM J Numer Anal 1985;22(5):1014–27.
[52] Shampine LF, Reichelt MW. The MATLAB ODE suite. SIAM J Sci Comput 1997;18(1):1–22.
[53] Shirayama S. Processing of computed vector fields for visualization. J Comput Phys 1993;106:30–41.
[54] Strid T, Rizzi A, Oppelstrup J. Development and use of some flow visualization algorithms. Lecture Series/Von Karman Institute for Fluid Dynamics 1989(7):1–56.
[55] Thiele MR. Streamline simulation. In: Proceedings of the sixth international forum on reservoir simulation, Fuschl, Austria, 2001.
[56] Thiele MR, Batycky RP, Blunt MJ. A streamline-based 3D field-scale compositional reservoir simulator, SPE 38889. In: Proceedings of the SPE annual technical conference and exhibition, San Antonio, Texas, 5–8 October 1997.
[57] Thiele MR, Batycky RP, Iding M, Blunt M. Extension of streamline-based dual porosity flow simulation to realistic geology. In: Proceedings of the 9th European conference on the mathematics of oil recovery, Cannes, France, 2004.
[58] Thiele MR, Batycky RP, Thomas LK. Miscible WAG simulations using streamlines. In: Proceedings of the 8th European conference on the mathematics of oil recovery, Freiberg, Germany, 2002.
[59] Verma S, Aziz K. A control-volume scheme for flexible grids in reservoir simulation, SPE 37999. In SPE reservoir simulation symposium, Dallas, Texas, 8–11 June 1997.
[60] Zheng C, Bennett GD. Applied contaminant transport modeling. 2nd ed. New York: Wiley-Interscience; 2002.

Paper B

On reproducing uniform flow exactly on general hexahedral cells using one degree of freedom per surface

*

* In *Advances in Water Resources*,doi:10.1016/j.advwatres.2008.11.005, 2008

Advances in Water Resources xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

On reproducing uniform flow exactly on general hexahedral cells using one degree of freedom per surface

J.M. Nordbotten [a,b], H. Hægland [b,*]

[a] Department of Civil and Environmental Engineering, Princeton University, NJ, USA
[b] Department of Mathematics, University of Bergen, Norway

ARTICLE INFO

Article history:
Received 16 June 2008
Received in revised form 12 November 2008
Accepted 19 November 2008
Available online xxxx

Keywords:
Uniform flow
Hexahedral cells
H(div)
Streamline tracing
Mixed finite element methods

ABSTRACT

The solution of second order elliptic partial differential equations typically arising for flow problems, e.g., in porous media, is frequently expressed as face fluxes of a flow field at hexahedral cells. In this letter, we show by example that the following properties are incompatible: (1) A local velocity reconstruction, (2) reproduction of uniform flow for general hexahedral cells, and (3) a flow field in *H*(div). This is in particular relevant to mixed finite element methods approximating *H*(div), and to other methods trying to reconstruct a flow field in a hexahedron using only local face fluxes, e.g., streamline tracing methods for reservoir simulation.

1. Introduction

Thirty years after Raviart and Thomas [29] introduced the mixed finite element method on rectangular meshes for numerical solution of second order elliptic partial differential equations, applications to distorted hexahedral grids remain challenging. Even if mixed methods using triangular or rectangular elements in 2D, or tetrahedral, parallelepiped, or prismatic elements in 3D have shown to be efficient and perform well, geophysical applications often require the flexibility of a general hexahedral mesh.

More generally, consider any method trying to build (interpolate) a flow field on a general hexahedron based on six boundary face fluxes. A key issue is then: Is it possible to obtain a flow field that both reproduce uniform flow and has point-wise continuity in the normal component across the boundary faces? In addition to mixed finite element methods, this problem is relevant to both streamline tracing [18] and control-volume mixed finite element methods [24].

To explain the reason for this difficulty, we start by reviewing the mixed finite element method for second order elliptic equations. With the aid of Lagrangian multipliers, the original partial differential equation is turned into a saddle point problem in variational form. The mixed method solves for a scalar field and a vector field (flow field) simultaneously. In order to prove that the approximate solution is unique and converges to the true solution, the vector field and scalar field must have certain regularities, i.e., belong to certain vector spaces.

A common choice for the vector field is the space [10]

$$H(\mathrm{div},\ \Omega) \equiv \{\boldsymbol{q} \in (L^2(\Omega))^3 : \nabla \cdot \boldsymbol{q} \in L^2(\Omega)\}.$$

If the discrete solution is sought in a subspace of the space of the true solution, the approximation is called conforming. In this case, the convergence to the true solution can be shown [10,14]. Below we will review conforming approximations of *H*(div). Nonconforming approximations can also be applied, but is less common, since it is more difficult to prove stability of the numerical method [15]. For other applications requiring approximations in *H*(div), see, e.g., [6,10,28,38].

Raviart and Thomas [29] and Thomas [36] considered second order elliptic partial differential equations, with the Poisson equation as a model problem. They developed a mixed finite element method for rectangles conforming in *H*(div), the so-called lowest order Raviart–Thomas elements (RT_0-elements), using the Hellinger–Reissner variational principle from elasticity. Nédélec [25] considered *H*(div) conforming extensions of these elements to three dimensions on rectangular parallelepipeds with application to the Maxwell's equations. These elements are commonly denoted RTN-elements. Brezzi et al. [9] improved on the RT-elements and developed the BDM family of mixed finite elements on rectangles and Nédélec [26] introduced a new family of mixed finite elements in 3D for rectangular parallelepipeds conforming in *H*(div) with application to the Stokes' equation. Brezzi et al. [7] developed

* Corresponding author.
E-mail addresses: jan.nordbotten@math.uib.no (J.M. Nordbotten), hakon.hagland@math.uib.no (H. Hægland).

0309-1708/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advwatres.2008.11.005

simplifications of the RT-elements on rectangular parallelepipeds, called the BDDF elements. Brezzi et al. [8] considered new versions of the BDM and BDDF elements. Brezzi and Fortin [10] and Roberts and Thomas [31] discuss $H(\mathrm{div})$ approximations and the relations between the RT, RTN, BDM, and BDDF elements. Later, Douglas and Wang [17] developed a new family of spaces for rectangular elements, the DW elements, and Chen [13] derived a new mixed formulation based on the BDM-elements.

For distorted elements, subspaces of $H(\mathrm{div})$ are constructed from a space of reference shape functions on a unit element [36]. Shen [34] and Wang and Mathew [37] seemed to be the first to consider general convex quadrilateral grids. Kuznetsov and Repin [21] introduced a composite mixed finite element for convex quadrilaterals, in which the quadrilateral was subdivided into two triangles. The space of functions corresponding to the subdivided quadrilateral was the space of $H(\mathrm{div})$-functions on the quadrilateral such that the restriction to each of the two triangles was in RTN_0 of the triangle and such that the divergence of the function was constant over the entire quadrilateral. Arnold et al. [2] have also considered mixed finite elements for convex quadrilaterals.

The problem of developing mixed finite elements for irregular hexahedral grids have been addressed by many authors in the last years. It is well known that the lowest order RT-elements on general hexahedral grids do not converge, see, e.g. [3,4,11,19]. This is related to the fact that the lowest order RT functions cannot represent the constant vector field on a general hexahedron [18,24,32]. Convergence in $H(\mathrm{div})$ can still be obtained for hexahedrons with flat surfaces [32] or grids which asymptotically tend to a parallelepiped mesh [3]. Kuznetsov and Repin [22] extended the idea of [21] to elements that are three-dimensional polygons with flat surfaces. Sboui et al.[32] developed a mixed finite element method on three-dimensional general hexahedral meshes with flat surfaces. The method was composite, meaning that each convex hexahedron was subdivided into five tetrahedra. Boarkine et al. [5] consider a mixed finite elements on non-matching distorted hexahedral meshes. They perform a fault interface reconstruction algorithm in order to construct a conforming polyhedral mesh in the whole domain and then apply the method of [22].

The problem with general hexahedral meshes is not restricted to the mixed finite element method. The control-volume mixed finite element method [12] also needs to define shape functions based on six face fluxes on general hexahedral meshes, see [24]. The last reference clearly shows the difficulty of treating general hexahedral meshes. In addition, many other locally conservative methods are also closely related to the mixed finite element method [20,39]. Note that recent research by Brezzi et al. [11] shows promising results on hexahedral grids for the mimetic finite difference method [33]. Other methods which rely on velocity constructions to obtain accurate flow results, include references [1,23,30]. One example where this local reconstruction of the velocity is needed, is streamline generation. Streamlines are used, e.g., in computational fluid dynamics for visualization of vector fields [27], in ground water flow for simulation of advective transport [35,40], and in hydrocarbon reservoir simulation [16].

A streamline tracing algorithm should satisfy certain tracing quality criteria, e.g., no crossing of streamlines, independence of starting point, reproduction of a uniform flow field, reproduction of symmetry, no premature termination of streamlines in cells or at cell boundaries, good approximation to the true solution, and computational efficiency. The importance of reproduction of uniform flow for the convergence of a streamline method, was shown in [18]. In this note, we also demonstrate the importance for a streamline method to use a flow field which lies in $H(\mathrm{div})$.

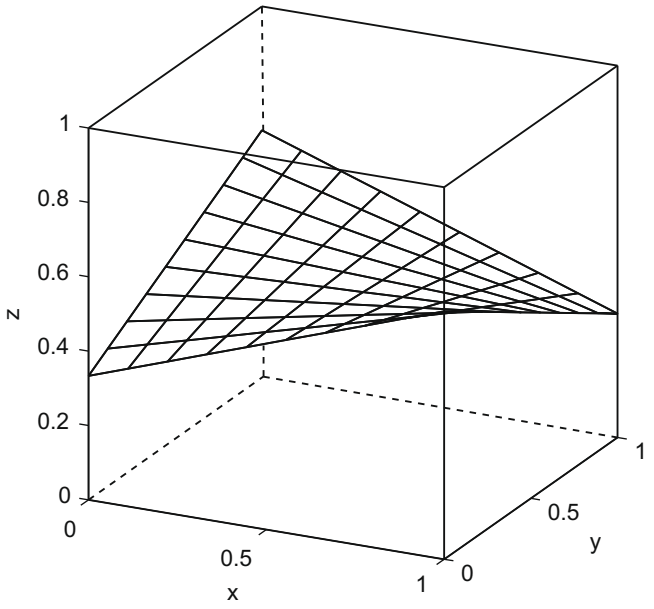


Fig. 1. A curved surface in a cubic domain.

2. Hexahedrons with curved surfaces

We will now, by construction of a counter-example, prove the following:

Theorem. *The following properties are incompatible, i.e., they cannot all be satisfied at the same time:*

(1) *A local velocity reconstruction based on total cell face fluxes.*
(2) *Reproduction of uniform flow for general hexahedral cells.*
(3) *A flow field in H(div).*

Note that the properties can in fact be satisfied for a subclass of problems, e.g., for hexahedrons with flat surfaces, or for uniform flow fluxes. What we show, however, is that they cannot be satisfied in general. Our findings are consistent with, and significantly generalize those of, [24].

Proof. We will assume that properties (1) and (2) hold, and then show, by example that property (3) cannot hold in general. Consider the cubic domain, $\Omega = [0,1]^3$. Discretize this domain with two cells, such that the first cell Ω_1 has four corners coinciding with the corners of Ω where $z = 0$, while the remaining corners are given by (0,0,1/3), (0,1,2/3), (1,1,1/3), and (1,0,2/3). The second cell, Ω_2, fills the rest of the domain such that the grid is a conforming hexahedral grid, with one internal curved surface, see Fig. 1.

Let $\boldsymbol{e}_x = [1,0,0]^T$. For unit flow $\boldsymbol{q} = \boldsymbol{e}_x$ parallel to the x-axis, the flux through the surfaces of Ω perpendicular to $\boldsymbol{e}_x$ will be 1. Thus the surfaces of Ω_1 and Ω_2 perpendicular to $\boldsymbol{e}_x$ have fluxes equal to 1/2. All other faces have zero integral flux, although the curved internal surface will have point-wise nonzero normal fluxes through the surface at all points except (1,1,1)/2.

Clearly, for our domain and flow field, a method satisfying property (2) will give correct, linear, velocity reconstruction. Property (1) ensures that this reconstruction is done individually for each cell. For this case, neither the flow field, nor the reconstructed field have any divergence, thus property (3) is not violated.

We now modify the example, by applying different flux boundary conditions on the surfaces perpendicular to $\boldsymbol{e}_x$. Let the lower cell, Ω_1, still have fluxes equal to 1/2, as in the previous

Please cite this article in press as: Nordbotten JM, Hægland H, On reproducing uniform flow exactly on general hexahedral cells using one ..., Adv Water Resour (2008), doi:10.1016/j.advwatres.2008.11.005

example. But for Ω_2, we impose fluxes equal to $-1/2$ at the faces perpendicular to $\boldsymbol{e}_x$. Due to mass conservation, the internal interface between Ω_1 and Ω_2 must have zero integral flux.

Note that, for Ω_1, all integral fluxes used in the velocity reconstruction is identical to our first example, thus the reconstructed flow field must be equal to the flow field $\boldsymbol{q}$ in the first example. However, for Ω_2, the fluxes now corresponds to a uniform flow field equal to $-\boldsymbol{q}$, and a method satisfying property (2) will reproduce this velocity.

Since the normal component of the velocity almost everywhere at the internal boundary is now non-zero and has opposite sign on each side of the boundary, the velocity field itself will not lie in H(div) in the neighborhood of the boundary. □

3. Remarks

In [24], it was shown that a local velocity field derived from the Piola-transformed RT_0 shape functions, or any 3D shape function that preserves a constant normal component across surfaces, cannot contain the constant vector fields on general hexahedra. Our findings are consistent with this observation, but more general, since it is valid for any local velocity interpolation using only the six local face fluxes as degrees of freedom.

In [18], a velocity interpolation scheme for general hexahedral grids was proposed for streamline tracing. The method was based on a corner point velocity interpolation (CVI), and it was shown to reproduce uniform flow on general hexahedral cells. Reproduction of uniform flow was shown to be essential for the convergence of streamlines and time-of-flight for nonuniformly refined grids.

The CVI method satisfied point-wise continuity in the normal component of the velocity across *planar* cell faces. However, the paper did not comment on the lack of continuity of the normal component for curved surfaces. The example of the previous section shows clearly that lack of normal continuity may cause prematurely termination of streamlines on cell boundaries. Hence, the conclusions of [18] is strictly only valid only for hexahedral cells with planar cell faces. Even if the numerical experiments in [18] revealed no problems for curved faces, the example of this note shows that there exist cases where the method will fail.

By taking into account fluxes in the six neighbouring cells of a hexahedron, thus relaxing property (1), it is possible that a modified CVI method satisfying properties (2) and (3) can be obtained. These ideas will be discussed in a separate paper.

4. Conclusion

We have shown by counter-example that the following three properties cannot be simultaneously satisfied: (1) a local velocity reconstruction based on integral cell face fluxes; (2) reproduction of uniform flow for general hexahedral cells; and (3) a flow field in $H(\mathrm{div})$. This result has consequences for streamline tracing and for mixed finite element methods attempting to approximate $H(\mathrm{div})$, and at the same time include the constant vector fields on general hexahedral meshes.

For streamline tracing, using a flow field which is not contained in $H(\mathrm{div})$ may cause streamlines to terminate incorrectly on cell boundaries. Whereas, failure to reproduce uniform flow, may cause convergence problems [18]. In this note, we have shown that a local flow field with degrees of freedom taken from six face fluxes, cannot achieve both reproduction of uniform flow and point-wise continuity in the normal component across cell boundaries, in general. The convergence of mixed finite element methods seems also to be closely related to the application of a flow field which both reproduce uniform flow and which lie in $H(\mathrm{div})$. We have shown that the latter properties cannot be obtained simultaneously for distorted hexahedrons and flow fields which are based six face fluxes.

Acknowledgements

The authors wish to thank Helge K. Dahle for inspiring discussions on topics related to velocity interpolations and their relationship to mixed finite element approximations.

References

[1] Aarnes JE, Kippe V, Lie KA. Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. Adv Water Resour 2005;28:257–71.
[2] Arnold DN, Boffi D, Falk RS. Quadrilateral *H*(div) finite elements. SIAM J Numer Anal 2005;42(6):2429–51.
[3] Bermúdez A, Gamallo P, Nogueiras MR, Rodríguez R. Approximation properties of lowest-order hexahedral Raviart–Thomas finite elements. CR Math 2005;340(9):687–92.
[4] Boffi D. On the finite element method on quadrilateral meshes. Appl Numer Math 2006;56:1271–82.
[5] Boiarkine OV, Kuznetsov YA, Svyatskiy DA. Diffusion equation on nonmatching distorted hexahedral meshes. Russ J Numer Anal Model 2007;22(4):311–24.
[6] Botha MM. Fully hierarchical divergence-conforming basis functions on tetrahedral cells, with applications. Int J Numer Meth Eng 2007;71(2):127–48.
[7] Brezzi F, Douglas Jr J, Durán R, Fortin M. Mixed finite elements for second order elliptic problems in three variables. Numer Math 1987;51:237–50.
[8] Brezzi F, Douglas Jr J, Fortin M, Marini LD. Efficient rectangular mixed finite elements in two and three space variables. RAIRO-Math Model Numer 1987;21(4):581–604.
[9] Brezzi F, Douglas Jr J, Marini LD. Two families of mixed finite elements for second order elliptic problems. Numer Math 1985;47:217–35.
[10] Brezzi F, Fortin M. Mixed and hybrid finite element methods. Springer series in computational mathematics, vol. 15. New York: Springer-Verlag; 1991.
[11] Brezzi F, Lipnikov K, Shashkov M, Simoncini V. A new discretization methodology for diffusion problems on generalized polyhedral meshes. Comput Meth Appl Mech Eng. 2007;196:3682–92.
[12] Cai Z, Jones JE, McCormick SF, Russell TF. Control-volume mixed finite element methods. Comput Geosci 1997;1:289–315.
[13] Chen Z. BDM mixed methods for a nonlinear elliptic problem. J Comput Appl Math 1994;53:207–23.
[14] Ciarlet PG. The finite element method for elliptic problems. Amsterdam: North-Holland; 1980.
[15] Croisille J-P, Greff I. Some nonconforming mixed box schemes for elliptic problems. Numer Meth Part D E 2002;18(3):355–73.
[16] Datta-Gupta A, King MJ. Streamline simulation: theory and practice. SPE Text book; 2007.
[17] Douglas J, Wang J. A new family of mixed finite element spaces over rectangles. Math Appl Comput 1993;12:183–97.
[18] Hægland H, Dahle HK, Eigestad GT, Lie K-A, Aavatsmark I. Improved streamlines and time-of-flight for streamline simulation on irregular grids. Adv Water Resour 2007. doi:10.1016/j.advwatres.2006.09.002.
[19] Jaffré J, Roberts JE, Sboui A. Mixed hexahedral finite elements for Darcy flow calculation. In: Proceedings of the XVI international conference on computational methods in water resources, June, Denmark, 2006.
[20] Klausen RA, Russell TF. Relationships among some locally conservative discretization methods which handle discontinuous coefficients. Comput Geosci 2004;8:341–77.
[21] Kuznetsov Y, Repin S. New mixed finite element method on polygonal and polyhedral meshes. Russ J Numer Anal Model 2003;18(3):261–78.
[22] Kuznetsov Y, Repin S. Convergence analysis and error estimates for mixed finite element method on distorted meshes. J Numer Math 2005;13(1):33–52.
[23] Maxwell RM, Welty C, Tompson FB. Streamline-based simulation of virus transport resulting from long term artificial recharge in a heterogeneous aquifer. Adv Water Resour 2003;26:1075–96.
[24] Naff RL, Russell TF, Wilson JD. Shape functions for velocity interpolation in general hexahedral cells. Comput Geosci 2002;6(3–4):285–314.
[25] Nédélec JC. Mixed finite elements in $\Re^3$. Numer Math 1980;35:315–41.
[26] Nédélec JC. A new family of mixed finite elements in $\Re^3$. Numer Math 1986;50:57–81.
[27] Nielson GM, Hagen H, Müller H, editors. Scientific visualization: overviews methodologies and techniques. Los Alamitos, Calif: IEEE Computer Society; 1997.
[28] Park EJ. Mixed finite element methods for generalized Forchheimer flow in porous media. Numer Meth Part D E 2005;21(2):213–28.
[29] Raviart PA, Thomas JM. A mixed finite element method for 2nd order elliptic problems. Lecture Notes in Mathematics 1977;606:292–315.
[30] Reichenberger V, Jakobs H, Bastian P, Helmig R. A mixed-dimensional finite volume method for two-phase flow in fractured porous media. Adv Water Resour 2006;29(7):1020–36.
[31] Roberts JE, Thomas J-M. Mixed and hybrid methods. In: Ciarlet PG, Lions JL, editors. Handbook of numerical analysis vol. II. Finite element methods (Part 1). Elsevier Science Publishers; 1991. p. 525–639.

Please cite this article in press as: Nordbotten JM, Hægland H, On reproducing uniform flow exactly on general hexahedral cells using one ..., Adv Water Resour (2008), doi:10.1016/j.advwatres.2008.11.005

[32] Sboui A, Jaffré J, Roberts JE. A composite mixed finite elements for general hexahedral grids. Technical report, INRIA, September 2007.
[33] Shashkov M. Conservative finite-difference methods on general grids. Symbolic and numeric computation series. Baca Raton: CRC Press; 1996.
[34] Shen J. Mixed finite element methods on distorted rectangular grids. Technical report. Texas A&M University; 1994.
[35] Strack ODL. Groundwater mechanics. Englewood Cliffs, NJ: Prentice Hall; 1989.
[36] Thomas JM. Sur l'analyse numerique des methodes d'elements finis hybrides et mixtes. PhD thesis. Paris, France: Université Pierre et Marie Curie; 1977.
[37] Wang J, Mathew T. Mixed finite element method over quadrilaterals. In: Dimov IT, Sendov B, Vassilevski P, editors. Advances in numerical methods and applications: proceedings of the third international conference. River edge, NJ: World Scientific; 1994. p. 203–14.
[38] Wang J, Ye X. New finite element methods in computational fluid dynamics by $H(\mathrm{div})$ elements. SIAM J Numer Anal 2007;45(3):1269–86.
[39] Younes A, Ackerer P, Chavent G. From mixed finite elements to finite volumes for elliptic PDEs in two and three dimensions. Int J Numer Meth Eng 2004;59:365–88.
[40] Zheng C, Bennett GD. Applied contaminant transport modeling. 2nd ed. New York: Wiley-Interscience; 2002.

Please cite this article in press as: Nordbotten JM, Hægland H, On reproducing uniform flow exactly on general hexahedral cells using one ..., Adv Water Resour (2008), doi:10.1016/j.advwatres.2008.11.005

Paper C

Adaptive streamline tracing for streamline simulation on irregular grids

*

* Presented at *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, 2006

ADAPTIVE STREAMLINE TRACING FOR STREAMLINE SIMULATION ON IRREGULAR GRIDS

H. HÆGLAND[1], H.K. DAHLE[1], K.-A. LIE[3], AND G.T. EIGESTAD[1,2]

[1]Department of Applied Mathematics, University of Bergen, 5008 Bergen, Norway
[2]Center for Integrated Petroleum Research, University of Bergen, 5020 Bergen, Norway
[3]SINTEF ICT, Department of Applied Mathematics, P.O. Box 124 Blindern, No-0314 Oslo, Norway

Abstract

Streamline simulation relies on an efficient and accurate calculation of streamlines and time-of-flight coordinates (TOF). Computation of streamlines are commonly done on a cell-by-cell basis using flux interpolation in the semi-analytical Pollock's method. An alternative method is to use corner-point velocity interpolation in each grid-cell, which reproduces uniform flow in three spatial dimensions. For this method, numerical integration of streamlines is required. In previous work, streamlines for irregular hexahedrons have been obtained by numerical integration in a reference cube, which facilitates easy detection of cell boundaries. The disadvantage of this method is the computational cost of transforming the velocity from physical space to a reference element, involving multiple evaluations of the Jacobian matrix. In this work, we propose an adaptive method for integration of streamlines directly in physical space, which aims at higher computational efficiency by reducing the number of evaluations of the Jacobian matrix.

1. Introduction

For a given velocity field $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{x})$, a family of streamlines $\boldsymbol{s} = \boldsymbol{s}(\tau; \boldsymbol{x}_0)$ is defined by

$$\frac{d\boldsymbol{s}}{d\tau} = \boldsymbol{q}(\boldsymbol{x}), \qquad \boldsymbol{s}(0; \boldsymbol{x}_0) = \boldsymbol{x}_0, \tag{1}$$

where τ parameterizes the streamline passing through the starting point $\boldsymbol{x}_0$.

We consider streamline tracing in the context of streamline simulation of (multiphase) flow in hydrocarbon reservoirs [*Batycky*, 1997; *Bratvedt et al.*, 1993; *King and Datta-Gupta*, 1998]. Multiphase flow in porous media can be modelled by a set of partial differential equations. Using the so-called fractional flow formulation, the flow of e.g., two phases can be described by a parabolic equation for the fluid pressure and a (hyperbolic) transport equation for fluid saturations. The basis for any streamline method is a sequential splitting of the coupled system of pressure and saturation equations, in which one first fixes the saturation and solves the pressure equation. The velocity field, which is linked to the pressure through Darcy's law, is then used as a parameter while advancing the transport equation a given step forward in time. The new saturation field is then used as input parameter for a new pressure solution step, and so on.

Introducing the time-of-flight coordinate [*Batycky*, 1997] as an integral along a streamline,

$$\tau(x,y,z) = \int_{s_0}^{s} \frac{\phi}{|\boldsymbol{q}|} ds, \tag{2}$$

the multidimensional saturation equation can be transformed into a family of one-dimensional equations along streamlines, which can the be used to speed up the numerical solution of fluid transport. In this work we will consider how to compute streamlines and time-of-flight from the solution of the pressure equation. The pressure equation will be solved by a mass-conservative scheme, e.g. an MPFA (O-method) [*Aavatsmark*, 2002], providing continuous fluxes on grid cell edges. We consider applications on 2D quadrilateral grids, where a continuously defined velocity field is commonly obtained by interpolation of the quadrilateral edge-fluxes [*Jimenez et al.*, 2005; *Prévost et al.*, 2002]. The extension of the proposed method to 3D is indicated.

2. Streamline Tracing on Irregular Grids

2.1. **Introduction: Tracing on Cartesian Grids.** We start with a discussion of streamline tracing on 2D Cartesian grids, commonly referred to as Pollock's method [*Pollock*, 1988] in the literature. Pollock's method builds a streamline as a series of line segments that each crosses a grid cell in physical space. The segments are constructed such that the exit point of the streamline in one cell is the entrance point in the next cell. By introducing a coordinate transformation, each rectangular grid cell is transformed into a unit square. Linear interpolation of (scaled) edge fluxes is then used to define a velocity field:

$$\boldsymbol{q}^{\mathrm{I}}(x,y) \equiv \begin{bmatrix} F_{x0}(1-x) + F_{x1}x \\ F_{y0}(1-y) + F_{y1}y \end{bmatrix}, \qquad 0 \le x \le 1, \quad 0 \le y \le 1. \tag{3}$$

Having defined a velocity field, the streamline $\boldsymbol{s}(\tau) = [x(\tau), y(\tau)]$ (on the unit square) is found by integrating the system of ODEs in (1):

$$\begin{cases} \dfrac{dx}{dt} = q_x^{\mathrm{I}}(x), & x(0) = x_0, \\ \dfrac{dy}{dt} = q_y^{\mathrm{I}}(y), & y(0) = y_0, \end{cases} \tag{4}$$

where q_x^{I} and q_y^{I} are the x- and y-components of $\boldsymbol{q}^{\mathrm{I}}$. Since q_x^I depends only on x, and q_y^{I} depends only on y, the streamline can be found analytically [*Pollock*, 1988]. Finally, the streamline is mapped back to the original grid cell.

2.2. **Streamline and Velocity in Curvilinear Coordinates.** Irregular grids are commonly used in reservoir description, due to ease of discretization and improved mesh quality. We consider irregular quadrilaterals in 2D, which are images of a unit square under the bilinear transformation,

$$\boldsymbol{x}(\hat{\boldsymbol{x}}) \equiv \sum_{i=1}^{4} \boldsymbol{x}_i \phi_i(\hat{x}, \hat{y}), \qquad 0 \le \hat{x}, \hat{y} \le 1 \tag{5}$$

where $\boldsymbol{x}_i = [x_i, y_i]$, $i = 1, \ldots 4$, are the four corner points of the quadrilateral, $\hat{\boldsymbol{x}} = [\hat{x}, \hat{y}]$ is a point in the unit square (the reference space $\mathcal{R}$); $\boldsymbol{x}(\hat{\boldsymbol{x}}) = [x(\hat{x}, \hat{y}), y(\hat{x}, \hat{y})]$ is a point in the

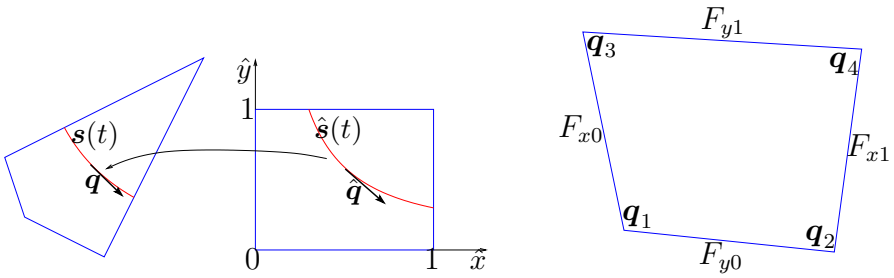


FIGURE 1. Left: Transformation of a streamline and velocity from reference space $\mathcal{R}$ to physical space $\mathcal{P}$. Right: Reconstructing velocities from fluxes

quadrilateral (physical space $\mathcal{P}$); and $\phi_i(\hat{x}, \hat{y})$, $i = 1, \ldots 4$, are the standard bilinear shape functions on the unit square. Note that the extension of the method to 3D will consider hexahedral grid cells that are images of a unit cube under a trilinear transformation.

Velocity Transformation. Commonly, streamlines are integrated (traced) in $\mathcal{R}$ and then mapped back to $\mathcal{P}$. To integrate a streamline $\hat{\boldsymbol{s}}(t)$ in $\mathcal{R}$, the velocity must be given in bilinear coordinates. An expression for this velocity may be derived as follows: To obtain the streamline $\boldsymbol{s}(t)$ in $\mathcal{P}$, the bilinear transformation is applied to the streamline $\hat{\boldsymbol{s}}(t)$ in $\mathcal{R}$; see Figure 1 (left). We then use the chain rule to determine the velocity in $\mathcal{R}$,

$$\boldsymbol{q} \equiv \frac{d\boldsymbol{s}}{dt} = \frac{d\boldsymbol{x}(\hat{\boldsymbol{s}}(t))}{dt} = \frac{d\boldsymbol{x}}{d\hat{\boldsymbol{x}}}\frac{d\hat{\boldsymbol{s}}}{dt} = \boldsymbol{J}\hat{\boldsymbol{q}}. \tag{6}$$

Here $\hat{\boldsymbol{q}} = d\hat{\boldsymbol{s}}/dt$ is the velocity in $\mathcal{R}$, $\boldsymbol{q}$ is the velocity in $\mathcal{P}$, and $\boldsymbol{J} = d\boldsymbol{x}/d\hat{\boldsymbol{x}}$ is the Jacobian matrix of the transformation. Thus, the transformed velocity is given by

$$\hat{\boldsymbol{q}} = \boldsymbol{J}^{-1}\boldsymbol{q}. \tag{7}$$

3. Corner Velocity Interpolation

The standard methods [*Cordes and Kinzelbach*, 1992; *Jimenez et al.*, 2005; *Prévost et al.*, 2002] for streamline tracing on irregular grids are based on integration in $\mathcal{R}$, approximating $\hat{\boldsymbol{q}}$ in Eq. (7) using a linear interpolation of the edge fluxes. In [*Hægland et al.*, 2005] it was shown that these methods may fail to reproduce uniform flow on irregular grids in 3D. To remedy this situation, a velocity interpolation scheme denoted *corner velocity interpolation* (CVI) was introduced. A central idea of this method is the shift from flux interpolation to corner-point velocity interpolation.

Reconstruction of Corner Velocities. The method is described here for the 2D case, see [*Hægland et al.*, 2005] for details of the 3D case. Consider the cell shown in Figure 1 (right), where four fluxes F_i, are given on the edges E_i, $i = x0, x1, y0, y1$. Normal vectors $\boldsymbol{n}_i$ with direction consistent with the sign of the edge fluxes are defined for each edge E_i. The length (absolute value) of the normal vectors are assumed to be equal to the length of the corresponding edge. Corner velocities $\boldsymbol{q}_i$, $i = 1, \ldots, 4$, at the corners $\boldsymbol{x}_i$, can then be obtained by solving 2×2 linear systems on the form

$$\begin{cases} \boldsymbol{q}_i \cdot \boldsymbol{n}_{x(i)} = F_{x(i)}, \\ \boldsymbol{q}_i \cdot \boldsymbol{n}_{y(i)} = F_{y(i)}, \end{cases} \qquad i = 1, \ldots, 4, \tag{8}$$

where $\{x(i)\}_{i=1}^4 = \{x0, x1, x1, x0\}$ and $\{y(i)\}_{i=1}^4 = \{y0, y0, y1, y1\}$, refer to edges in the x- and y-direction, respectively, adjacent to corner $\boldsymbol{x}_i$. This means that to determine $\boldsymbol{q}_1$, we solve

$$\begin{cases} \boldsymbol{q}_1 \cdot \boldsymbol{n}_{x0} = F_{x0}, \\ \boldsymbol{q}_1 \cdot \boldsymbol{n}_{y0} = F_{y0}, \end{cases} \tag{9}$$

since E_{x0} and E_{y0} are adjacent to corner $\boldsymbol{x}_1$. The systems (8) are well-conditioned as long as the quadrilateral does not degenerate. If the fluxes have been computed exactly for a uniform flow field $\boldsymbol{q}$, then $\boldsymbol{q}_i = \boldsymbol{q}$. See *Hægland et al.* [2005] for more details.

The CVIR Method. The method of *corner velocity interpolation in reference space* (CVIR) is based on integration of (7) in $\mathcal{R}$, where the unknown quantity is the velocity $\boldsymbol{q}$ in $\mathcal{P}$. We approximate $\boldsymbol{q}$ by a bilinear interpolation of the corner velocities $\boldsymbol{q}_i$ in Equation (8),

$$\boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVIP}}(\boldsymbol{x}(\hat{x}, \hat{y})) \equiv \sum_{i=1}^{4} \boldsymbol{q}_i \phi_i(\hat{x}, \hat{y}), \tag{10}$$

where ϕ_i and $\boldsymbol{x}$ have the same meaning as in Eq. (5). Note that (10) is exact for uniform flow. The method of *corner velocity interpolation in physical space* (CVIP) to be described in the next section, integrates the velocity in (10) directly in physical space, hence the subscript CVIP in (10). The difficulty with this approach is to determine when cell boundaries are crossed, especially in 3D. Thus, integration on a reference element was suggested in *Hægland et al.* [2005], leading to the CVIR method characterized by the following velocity interpolation in $\mathcal{R}$,

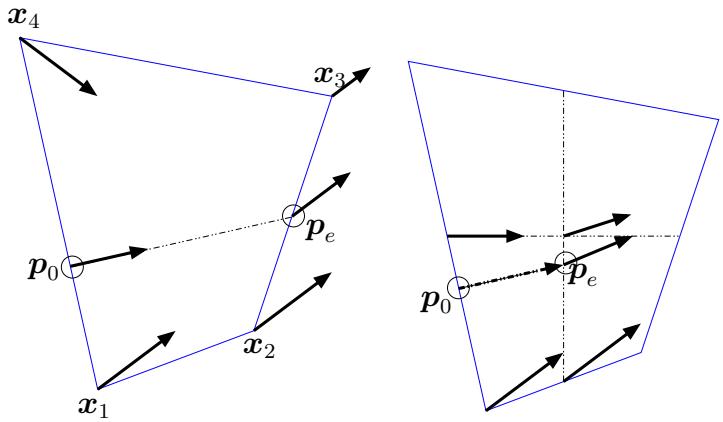
$$\hat{\boldsymbol{q}}^{\mathrm{I}}_{\mathrm{CVIR}} \equiv \boldsymbol{J}^{-1} \boldsymbol{q}^{\mathrm{I}}_{\mathrm{CVIP}}, \tag{11}$$

where we have used (7). The disadvantage of the CVIR-method is the expensive evaluation of the inverse Jacobian matrix at each integration point.

4. Tracing in Physical Space. The CVIP Method

In [*Hægland et al.*, 2005], a forth-order Runge-Kutta method (RK45) was used for integration with the CVIR method, which requires six function evaluations (i.e. evaluations of velocity) per time step. For each velocity evaluation we need to determine the inverse of the Jacobian, see (11). In order to avoid this expensive step, we would like to consider integration in the physical space. Integration in $\mathcal{P}$ involves the difficulty of determining the exit point from a cell, additionally, if we use a Runge-Kutta method, function evaluations (velocity) at intermediate points in physical coordinates are needed, whereas the velocity at these points is given in reference coordinates, see (10). This means that an inverse bilinear transformation must be used to determine the velocity at these points, which is even more expensive than evaluating the inverse Jacobian matrix.

In this paper we instead apply an Euler predictor-corrector method for integration in $\mathcal{P}$, which only requires the velocity at points on the cell boundaries. At such points, the inverse bilinear transformation will simplify to a linear relation, and the velocity in (10) can be easily evaluated. Adaptivity is implemented using a recursive mesh refinement procedure. Similar ideas have been explored in [*Bensabat et al.*, 2000; *Cheng et al.*, 1996]. Consider the quadrilateral grid cell shown in Figure 2 with reconstructed corner velocities $\boldsymbol{q}_i$ at the corners $\boldsymbol{x}_i$, $i = 1, \ldots, 4$, respectively. Within the cell, the velocity $\boldsymbol{q}$ is given



(a) Original grid cell (b) Refined grid cell

FIGURE 2. Criterion for grid refinement

by the bilinear interpolation in (10). A particle enters the cell at $\boldsymbol{p}_0$ with velocity $\boldsymbol{v}_0$ (determined by the interpolation). If the flow field is uniform, the streamline through the cell will be a straight line, as indicated in the figure. An approach to check the validity of the assumption of uniform velocity is to compare the velocity $\boldsymbol{v}_e$ at the exit point $\boldsymbol{p}_e$ with $\boldsymbol{v}_0$. Accordingly, we define the relative velocity error as

$$E_v = \frac{\|\boldsymbol{v}_0 - \boldsymbol{v}_e\|}{\|\boldsymbol{v}_0\|} \tag{12}$$

If E_v is less than some prescribed tolerance ϵ, we accept the constant approximation $\boldsymbol{q} \approx \boldsymbol{v}_0$ (Euler method), or we rather use the improved estimate $\boldsymbol{q} \approx (1/2)(\boldsymbol{v}_0 + \boldsymbol{v}_e)$ (Euler predictor-corrector). If E_v is not less than ϵ, i.e., $\epsilon < E_v < E_{\max}$, where $E_{\max}$ is a prescribed maximum error, an adaptive refinement of the grid cell will be employed. If E_v is greater than $E_{\max}$, we expect a high degree of refinement, and it would probably be more efficient to apply the RK45 method in reference space.

Mesh Refinement. The mesh refinement is based on a recursive splitting of a cell into four sub-cells. The splitting is defined by the coordinates of the bilinear transformation for a splitting of the the unit square in $\mathcal{R}$ into four sub-squares of equal size, see Figure 2(b). If E_v exceeds ϵ for a given sub-cell, that sub-cell is recursively split into four new sub-cells. Note that the remaining three sub-cells may be left unrefined by this procedure.

5. Numerical Experiments

To illustrate the method we consider numerical experiments in 2D. The extension to 3D is work in progress.

5.1. **A Synthetic Case.** Consider the test case in Figure 3. The domain has unit permeability, except in the cross-hatched area, where $K = 10^{-3}$. The boundary conditions are of Neumann type, with no-flow on top and bottom, inflow at edge AC, and outflow at edge BD. A single-phase pressure equation is solved using an MPFA-method on an irregular grid with 10×10 cells to provide continuous fluxes on each grid cell edge. The irregular grid is obtained from a Cartesian uniform grid, where each cell is a square with

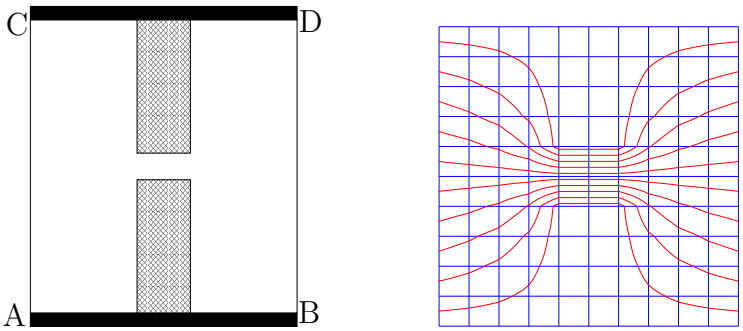


FIGURE 3. Domain used in the first numerical test case. Streamlines for a Cartesian grid of the domain are shown to the right.

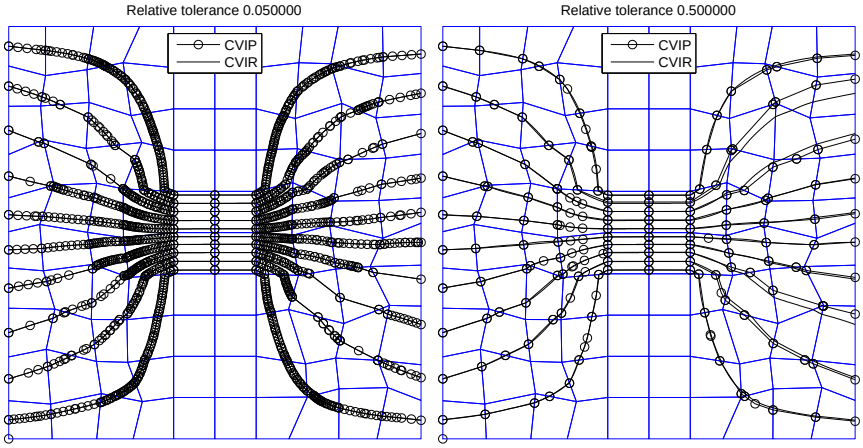


FIGURE 4. Streamlines for the first test case. Left: $\epsilon = 0.05$. Right: $\epsilon = 0.5$.

side length s_u. In order to keep the physical properties of the domain the same for different grid perturbations, only grid points in the interior of the left and right parts of the grid are perturbed. These points are perturbed within squares with side length s_p centered at the original grid point. For a p% perturbation we have $100 s_p/s_u = p$, see Figure 4, where we have used a 50% perturbation.

Test 1: 10 Streamlines. Corner velocities for each grid cell are obtained from the fluxes using the recovery method from Section 3. For each integration method (CVIP/CVIR), ten streamlines are traced from ten fix points on the inflow boundary until they exit the domain at edge BD, see Figure 4. For the CVIR method, a fourth-order Runge-Kutta method with relative tolerance of 10^{-8} is used for integration in $\mathcal{R}$. Thus the streamlines traced with the CVIR method can be viewed as reference solutions for the given interpolated velocity field. Streamlines for CVIR and for CVIP with $\epsilon = 0.05$ and $\epsilon = 0.5$ are shown in Figure 4. The circles indicate the length of each local step in the CVIP method. We see the benefit of using the CVIP method for uniform flow (the middle of the domain in Figure 4), where no grid refinement is needed and the method is also exact. The influence of the streamline launching point on errors in streamline shape and time-of-flight has been investigated by *Matringe and Gerritsen* [2004]. They found that

streamlines should be started as far away from singularities (wells/fractures) as possible. In our test case we have a region with high streamline density (the fracture in the middle of the figure) and two regions with low streamline density (the inflow boundary and the outflow boundary). If we start the streamlines at the inflow boundary they will converge into the fracture, which will reduce errors. On the other hand, as the streamlines exit the fracture, they will diverge to the outflow boundary and the errors will increase, see Figure 4 (right). Still, it would be worse to launch the streamlines in the fracture and trace forward and backwards: For this case we then get the same errors at outflow boundary, but in addition we get errors at inflow boundary of the same size as for the outflow boundary.

Test 2: 100 Streamlines. We consider a sequence of relative tolerances $\{\epsilon_{i=1}^{8}\} = \{0.005, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64\}$. We increase the number of streamlines to 100, still starting at equally spaced points at the inflow boundary, and measure the error in the CVIP method at the right boundary BD compared to the reference solution given by the CVIR method. Two different error measures are applied: error in the exit position and error in time-of-flight at the exit point. We consider a sequence of grids with perturbations $\{p_{i=1}^{6}\} = \{0, 10, 20, 30, 40, 50\}$. For each p_i and each ϵ_i, we take the average of the errors over the 100 streamlines. The resulting position errors are shown in Figure 5 (left). The

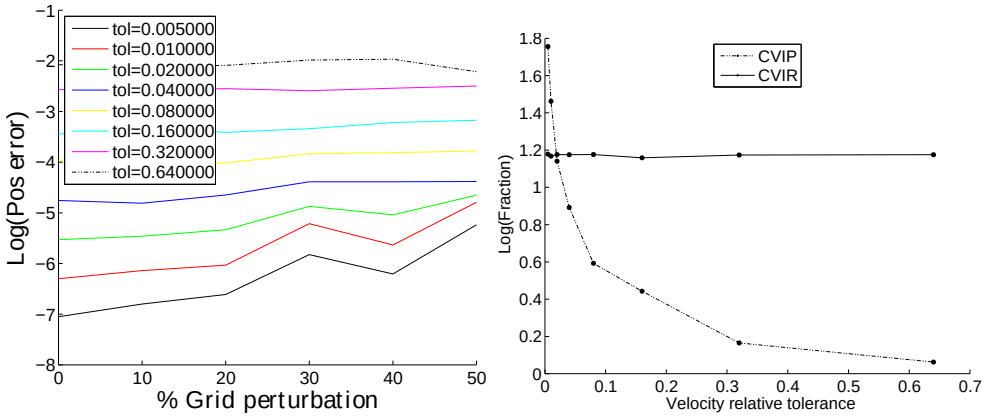


FIGURE 5. Left: Error in position. Right: Runtime for CVIP for different values of ϵ.

results for the time-of-flight error look similar. As expected, the errors are not sensitive to grid perturbation and increase with increased relative tolerance. Finally, we compare the runtime required for the CVIP and CVIR methods to trace all 100 streamlines for a given ϵ. We measure the runtime relative to the standard flux mapping (SFM) described in [*Prévost et al.*, 2002]. This is a semi-analytical method for streamline tracing, and is commonly used in the industry due to its speed. Note that the extended flux mapping (EFM) by *Jimenez et al.* [2005], which is able to reproduce time-of-flight for uniform flow in 2D, is about as fast as the SFM method. The runtime for the CVIR and CVIP will be given in fractions of the runtime of the SFM method. For a MATLAB implementation, results for base-ten logarithm of the fraction for different tolerances ϵ_i, are shown in Figure 5 (right). A value of zero corresponds to the same runtime as the SFM. The

runtime of the CVIR method is approximately 15 times as long as the runtime of the SFM method. We see that the runtime of the CVIP method decreases as ϵ increases. The CVIP method is faster than the CVIR method for $\epsilon > 0.04$. For this specific case, there is a large reduction in runtime when going from $\epsilon = 0.04$ to approximately $\epsilon = 0.32$, where the CVIP method is about 12 times as fast as the CVIR method. In this region of ϵ, the position error also increases slowly. For $\epsilon > 0.32$ the opposite is true: there is not so much to gain in runtime, whereas the error increases more rapidly. So a choice of $\epsilon \approx 0.3$ could be reasonable in order to balance accuracy and speed. In other situation, other considerations might apply; for instance, for the special case of uniform flow, any ϵ might be used.

5.2. **The SPE 10 case.** This test case is taken from the tenth SPE comparative solution project, Model 2, see http://www.spe.org/csp/datasets/set02.htm. The original fine scale model is a Cartesian grid of $60 \times 220 \times 85$ grid cells.

Test 1: Layer 14. To set up a 2D simulation, we pull out layer number 14 and ignore the z-coordinates. Each interior grid point of the resulting grid is perturbed slightly (20%) to get an irregular grid. The original permeability is left unchanged in each perturbed cell. No-flow boundary conditions are applied, with an injector in the lower left corner and a producer in the upper right corner, see Figure 6. Streamlines are then initiated at the injector cell and traced to the producer. We consider four streamline starting points

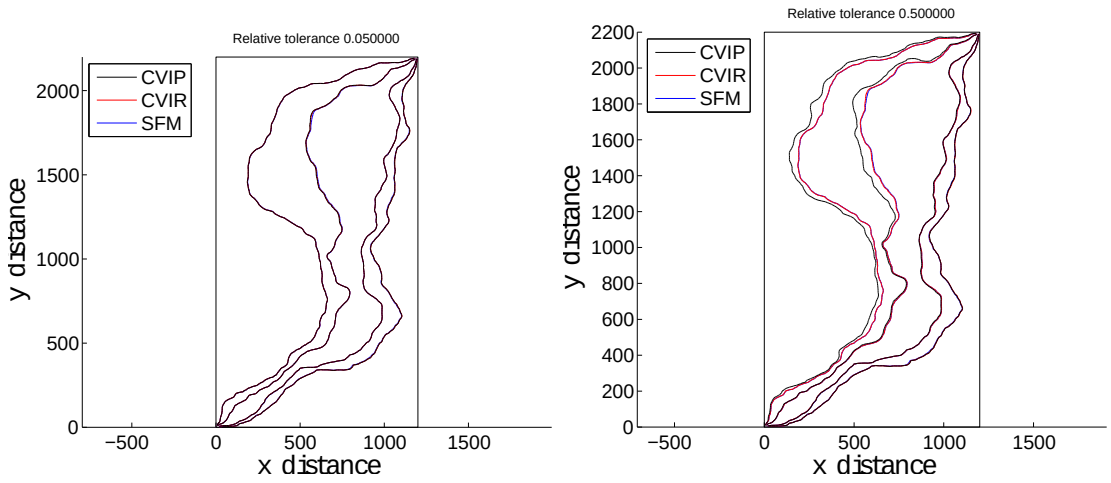


FIGURE 6. SPE 10 case. CVIP with $\epsilon = 0.05$ (Left). CVIP with $\epsilon = 0.5$ (Right)

at the injector cell. For each starting point we trace the streamline with three different methods: CVIR, CVIP, and SFM. The only difference between the two plots in Figure 6, is the relative tolerance used with the CVIP method. For $\epsilon = 0.5$ in Figure 6 (right), we get large errors in the CVIP method compared to the CVIR method (and the SFM method). For $\epsilon = 0.05$ in Figure 6 (left), there is hardly any difference between the methods. In Table 1 the time-of-flight at the producer is shown for all the methods.

TABLE 1. Time-of-flight at the producer for the different methods

	SL1	SL2	SL3	SL4
SFM	1.8016e+06	2.1706e+06	1.6297e+06	1.3567e+06
CVIR	1.7990e+06	2.2308e+06	1.6556e+06	1.3584e+06
CVIP ($\epsilon = 0.05$)	1.7985e+06	2.2613e+06	1.6562e+06	1.3575e+06
CVIP ($\epsilon = 0.5$)	2.2433e+06	2.5604e+06	1.6655e+06	1.3842e+06

Test 2: All Layers. For each layer in the SPE case we consider the same setup as in the previous test. We increase the number of streamlines to ten, and trace from equally distributed starting points on the two injector faces. Let τ_R and τ_P be the time-of-flight at the producer for a particular streamline computed by CVIR and CVIP, respectively. Then the error for that streamline is measured as $|\tau_R - \tau_P|/\tau_R$. For each layer we take the average over all ten streamlines. The base-ten logarithm of the resulting error is shown in Figure 7 (left) for different values of ϵ. In Figure 7 (right) we have plotted the

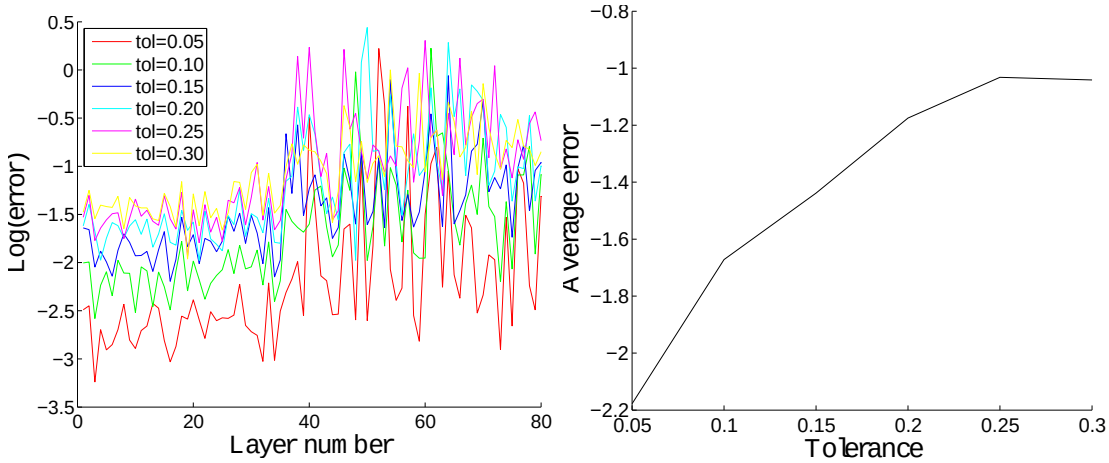


FIGURE 7. SPE 10 case. Left: Error in CVIP for different layers and different ϵ. Right: Average error of CVIP over all layers for different ϵ.

average of the errors over all layers. A value of -2 corresponds to a 1% relative error in the time-of-flight. This is achieved on average for $\epsilon \approx 0.07$. The results show that the relative errors vary from 100% to 0.1% for different layers, streamlines, and tolerances ϵ. The right figure shows that the average error increases with ϵ, as expected. We believe that the high values for, e.g., $\epsilon = 0.05$ (red curve) in some layers (Figure 7 (left)) are not related to the tolerance, or the tracing method (CVIP). By following a particular streamline with a large error, we found that the error was a result of large variations in permeability, which for some grid cells caused the streamline to be extremely sensitive to the entry point.

5.3. **Conclusion.** We have tested the CVIP method on 2D irregular grids and compared it with the CVIR, SFM, and EFM methods. Experience with the 2D case can be valuable when extending the method to 3D. For uniform or almost uniform flow, CVIP is as efficient as the EFM method. Since the CVIP method is adaptive, it is possible to shift

to a more efficient and accurate method (e.g., the EFM, which is as accurate as CVIR in 2D) when the velocity field becomes less uniform. Similarly, in 3D one could make an adaptive method in which CVIP is the default choice and the more accurate CVIR is applied in regions where the velocity field changes rapidly. We have seen that if the tolerance of the CVIP method is chosen too high, meaning that we accept large errors in our approximation of the velocity, we are likely to get large errors in the streamlines. Accordingly, an error estimate for the CVIP method that can be used to control the tolerance, would be of great value.

References

Aavatsmark, I. (2002), An introduction to multipoint flux approxmations for quadrilateral grids, *Comput. Geosci.*, *6*, 405–432.

Batycky, R. (1997), A three-dimensional two-phase field scale streamline simulator. (http://www.streamsim.com/index.cfm?go=about.publications), Ph.D. thesis, Stanford University, Dept. of Petroleum Engineering.

Bensabat, J., Q. Zhou, and J. Bear (2000), An adaptive pathline-based particle-tracking algorithm for the eulerian-lagrangian method, *Adv. Water Resour.*, *23*, 383–397.

Bratvedt, F., K. Bratvedt, C. Buchholz, T. Gimse, H. Holden, L. Holden, and N. Risebro (1993), Frontline and frontsim: two full scale, two-phase, black oil reservoir simulators based on front tracking, *Surveys Math. Indust.*, *3*, 185–215.

Cheng, H.-P., J.-R. Cheng, and G.-T. Yeh (1996), A particle tracking technique for the lagrangian-eulerian finite element method in multi-dimensions, *Internat. J. Numer. Methods Engrg.*, *39*, 1115–1136.

Cordes, C., and W. Kinzelbach (1992), Continous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements, *Water Resour. Res.*, *28*(11), 2903–2911.

Hægland, H., H. Dahle, G. Eigestad, K.-A. Lie, and I. Aavatsmark (2005), Improved streamlines and time-of-flight for streamline simulation on irregular grids, submitted to *Advances in Water Resources*.

Jimenez, E., K. Sabir, A. Datta-Gupta, and M. King (2005), Spatial error and convergence in streamline simulation. SPE 92873, in *SPE Reservoir Simulation Symposium, Woodlands, Texas, 31 Jan. -2 Feb.*

King, M., and A. Datta-Gupta (1998), Streamline simulation: A current perspective, *In Situ*, *22*(1), 91–140.

Matringe, S., and M. Gerritsen (2004), On accurate tracing of streamlines, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 Sep.*

Pollock, D. (1988), Semi-analytical computation of path lines for finite-difference models, *Ground Water*, *26*(6), 743–750.

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

Paper D

Streamline methods on fault adapted grids for risk assessment of storage of CO_2 in geological formations

*

* Presented at *Proceedings of the XVI International Conference on Computational Methods in Water Resources*, 2006

STREAMLINE METHODS ON FAULT ADAPTED GRIDS FOR RISK ASSESSMENT OF STORAGE OF CO_2 IN GEOLOGICAL FORMATIONS

H. HÆGLAND[1], H.K. DAHLE[1], G.T. EIGESTAD[3,1], J.M. NORDBOTTEN[1], M.A. CELIA[2], AND A. ASSTEERAWATT[4]

[1]Department of Applied Mathematics, University of Bergen, 5008 Bergen, Norway
[2]Department of Civil and Environmental Engineering, Princeton University, Princeton 08544, NJ, USA
[3]Center for Integrated Petroleum Research, University of Bergen, 5020 Bergen, Norway
[4]University of Stuttgart, Institute of Hydraulic Engineering, Pfaffenwaldring 61, 70569 Stuttgart, Germany

Abstract

Geological storage of CO_2 in possibly fractured and faulted media, involves the risk of leakage. The extent of leakage may be assessed with statistical methods through analysis of simulations of multiple realizations of a stochastic model. Numerical simulation of numerous such realizations typically requires considerable computational cost, motivating the use of fast numerical methods, such as streamline simulation, for screening. Streamline methods have shown to be effective for reservoir characterization and simulation. In this work we will develop methodology which allows for tracing of streamlines in fractured or faulted media. The work is motivated in part by the need to assess potential of geological storage of CO_2 and is also highly relevant for reservoir simulation.

1. Introduction

Carbon dioxide (CO_2) storage is considered to be a potential key strategy to reduce anthropogenic CO_2 emissions. The principle of carbon dioxide storage is to capture CO_2 produced with the conversion of fossil fuels and sequester the CO_2 in a geological reservoir. Understanding of the risks (see [*Damen et al.*, 2004] for an overview) associated with CO_2 sequestration is one of the key factors affecting public acceptance. The main research topic in risk associated with underground CO_2 sequestration, is leakage. Locally, leakage of CO_2 may be dangerous (in elevated concentrations) to humans, animals and ecosystems, whereas globally, high leakage rates may render the sequestration of CO_2 ineffective as a mitigation option. Typically leakage occurs either as diffuse seepage through the caprock and subsequent overlaying formations, or as concentrated leakage through potentially highly conductive paths such as (abandoned) wells or fractures. From a risk assessment perspective, abandoned wells and fractures form a major challenge since their properties are often at best known statistically [*Celia et al.*, 2004].

Numerical simulation of CO_2 sequestration for risk assessment is typically done in a statistical framework, where evaluation and screening of multiple realizations of a model requires fast numerical methods. Streamline methods have shown to be effective for

reservoir characterization and simulation. In this work we will develop methodology which allows for tracing of streamlines in fractured or faulted media. The basis for a streamline method is a sequential splitting of the coupled pressure and saturation equations. A mass-conservative discretization, which handles general faulted grids in a consistent manner, will be used for the pressure equation. In earlier work we have developed streamline tracing on structured and unstructured matching grids. Here we present an extension to grids adapting to faulted media.

Governing equations. For simplicity, the model equations for CO_2 sequestration will be given for two-phase flow, neglecting gravity and capillary pressure. Inclusion of these effects has been discussed elsewhere, e.g. [*Gerritsen et al.*, 2005]. The pressure equation can then be written as [*Settari and Aziz*, 1972]

$$c_t \partial_t p + \nabla \cdot \boldsymbol{q} = b_p, \tag{1}$$

where $\boldsymbol{q}$ is the total velocity (sum of phase velocities), c_t is the total compressibility, and b_p is a source term. Equation (1) is linked to a transport equation for the fluid saturation S

$$\phi \partial_t S + \nabla \cdot \big(\boldsymbol{q} f(S, \boldsymbol{x})\big) = b_s, \tag{2}$$

through Darcy's equation for the volumetric flow density,

$$\boldsymbol{q} = -\lambda(S, \boldsymbol{x}) \nabla p. \tag{3}$$

Here, ϕ, λ, f, and b_s denote porosity, total mobility, fractional flow function, and source terms, respectively. By introducing the time-of-flight [*Batycky*, 1997; *Blunt et al.*, 1996] as an integral along a streamline,

$$\tau(x, y, z) = \int_{s_0}^{s} \frac{\phi}{|\boldsymbol{q}|} ds, \tag{4}$$

the saturation equation, Equation (2), reduces to the one-dimensional hyperbolic equation,

$$\partial_t S + \partial_\tau f(S) = b_s - f(S) \nabla \cdot \boldsymbol{q}. \tag{5}$$

The last term on the right-hand side accounts for compression or expansion of fluids in the case of compressible flows and is identically zero for incompressible flows. Solving the family of one-dimensional problems (5) on a discrete set of streamlines is often much faster than solving Equation (2) over a grid in physical space. Streamlines $\boldsymbol{s} = \boldsymbol{s}(\tau; \boldsymbol{x}_0)$ are defined for a given velocity field $\boldsymbol{q} = \boldsymbol{q}(\boldsymbol{x})$, by

$$\frac{d\boldsymbol{s}}{d\tau} = \boldsymbol{q}(\boldsymbol{x}), \qquad \boldsymbol{s}(0; \boldsymbol{x}_0) = \boldsymbol{x}_0, \tag{6}$$

where τ parameterizes the streamline passing through the starting point $\boldsymbol{x}_0$.

In this work, the domain will be discretized into an unstructured grid, capable of representing faults and fractures. We consider single-phase tracer flow in 2D, where an incompressible version of the pressure equation (1) will be solved on quadrilateral, possibly nonmatching grids, using an MPFA (O-method) [*Aavatsmark*, 2002]. The MPFA-method supplies continuous fluxes on each quadrilateral edge, and a continuously defined velocity field is obtained by interpolation of these edge-fluxes. Tracing of streamlines (integration of (6)) for use in flow simulations has been investigated in [*Hægland et al.*, 2005; *Jimenez*

et al., 2005; *King and Datta-Gupta*, 1998; *Matringe*, 2004; *Matringe and Gerritsen*, 2004; *Matringe et al.*, 2005; *Prévost*, 2003; *Prévost et al.*, 2002; *Sun et al.*, 2005].

2. Streamline tracing on faulted grids

2.1. **Introduction: Tracing on Cartesian grids.** To motivate the description of streamline tracing on faulted grids, we start by discussing the basic version on 2D Cartesian grids, which is commonly referred to as Pollock's method [*Pollock*, 1988]. Pollock's method builds a streamline as a series of line segments that each crosses a grid cell in physical space. The segments are constructed such that the exit point of the streamline in one cell is the entrance point in the next cell. By introducing a coordinate transformation, each rectangular grid cell is transformed into a unit square. Linear interpolation of (scaled) edge fluxes is then used to define a velocity field:

$$\boldsymbol{q}^{\mathrm{I}}(x,y) \equiv \begin{bmatrix} F_{x0}(1-x) + F_{x1}x \\ F_{y0}(1-y) + F_{y1}y \end{bmatrix}, \qquad 0 \le x \le 1, \quad 0 \le y \le 1. \tag{7}$$

2.2. **Tracing on faulted grids.**

Solving the pressure equation. For faulted 2D grids we will use a flux continuous control-volume method [*Aavatsmark et al.*, 2001] for the pressure equation. A discrete linear system for cell centre pressures is obtained by assuming a linear potential in each grid cell, and by requiring potential and flux continuity in interaction regions surrounding grid points. The solution procedure will automatically give continuous fluxes on half-edges of each grid cell.

Streamline tracing. Pollock's method cannot be immediately applied to faulted grid cells, since the input is half-edge fluxes. A simple approach may be to add the half-edge fluxes. However, this simple approach causes streamlines to terminate at a no-flow half-edge; see [*Jimenez et al.*, 2005]. This is because adding two half-edge fluxes where one is zero and the other nonzero, will produce a nonzero whole-edge flux, and the information about the non-flow boundary has been lost. A better approach is a subdivision of the cell such that the half-edge fluxes become whole-edge fluxes in the new cells. Mass conservation is used to determine the fluxes on the unknown edges [*Jimenez et al.*, 2005].

3. Streamline tracing for fractured media

Solving the pressure equation. A discretization of a fractured medium domain with volumetric elements in the fractures requires a grid which resolves the geometry of the problem. Due to the small fracture widths, the resulting grid will either consist of a very large number of grid cells or the grid will contain cells with a very large aspect ratio [*Reichenberger et al.*, 2006].

A solution to this is to apply a mixed-dimensional discretization method which realizes fractures as lower-dimensional elements. Solution of the pressure equation for such a model is described in [*Martin et al.*, 2005; *Reichenberger et al.*, 2006].

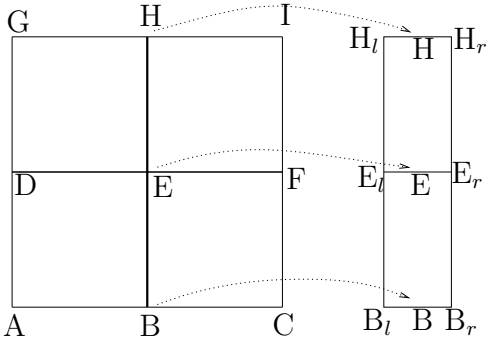
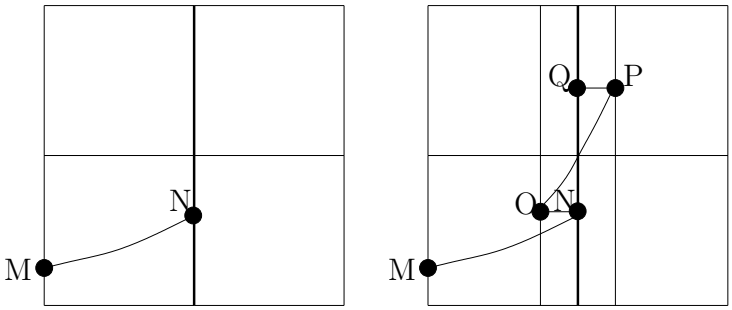


FIGURE 1. Section of a grid with a vertical fracture along (BH) (left). The fracture is associated with a 2D virtual element of width d for computational purposes (right).

Streamline tracing. In a 2D domain, a fracture will be modelled as a 1D object associated with a small width, see Figure 1. By the 2D fracture we will mean the 1D fracture associated with a small width, d, see Figure 1, where the size of d has been exaggerated for illustration purposes. Each edge of the 2D fracture will have an associated flux. Note that horizontal edges for the 1D fracture reduce to a single point in the left part of Figure 1. There is still a flux associated with these edges. Consider tracing a streamline for this simple grid, see Figure 2. A streamline reaching the point N in the fracture, is traced



(a) A streamline enters the grid at the point M, and reaches the fracture at point N.

(b) The streamline tracing through the fracture is done in the 2D fracture.

FIGURE 2. Streamline tracing through a fracture.

through the 1D fracture using the 2D fracture fluxes. The entry point N is associated with a starting point O. The streamline exits the 2D fracture at P, which is associated with a point Q in the 1D fracture. Points O and P are hence pictures in order to calculate exit/entry points of the fracture.

4. Tracer flow modelling

When acid CO_2 is dissolved in water prior to injection in an aquifer for sequestration [*Bachu and Gunter*, 2004], a one-phase miscible displacement will take place. We will

assume a simplified model of this displacement, where we consider two fluids, which are completely miscible. The amount of the first fluid contained in the mixture has no influence on the flow of the mixture, hence the name tracer. The volume fraction $C(\boldsymbol{x}, t)$ of the tracer is defined as [*Bastian*, 1999]

$$C(\boldsymbol{x}, t) = \frac{\text{volume of tracer in REV}}{\text{volume of mixture in REV}} \tag{8}$$

Further, we assume that the fluids have the same density ρ. The conservation of mass for tracer can then be modelled by

$$\frac{\partial(\phi\rho C)}{\partial t} + \nabla \cdot \{\rho \boldsymbol{q} C - D\nabla C\} = \rho q_T \tag{9}$$

where the velocity $\boldsymbol{q}$ is the Darcy velocity for the single phase, and D is the hydrodynamic dispersion tensor. Equation (9) is simplified by assuming incompressible flow and neglecting dispersion, which leads to

$$\phi\frac{\partial C}{\partial t} + \boldsymbol{q} \cdot \nabla C = 0 \tag{10}$$

away from sources and sinks.

We observe that the advection equation for the tracer resembles the saturation equation for two phase flow, i.e., Eq. (2). By introducing the time-of-flight relation (4), Eq. (10) simplifies to a one dimensional linear advection equation along the streamlines,

$$\frac{\partial C}{\partial t} + \frac{\partial C}{\partial \tau} = 0. \tag{11}$$

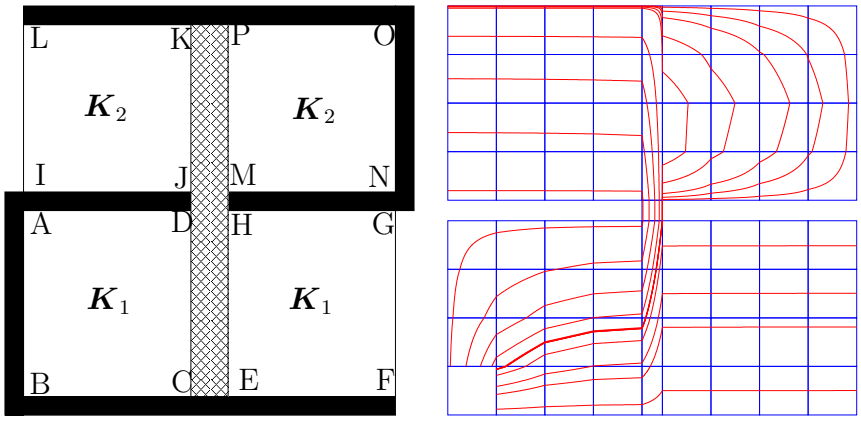
If we use the initial condition $C(0, \tau) = C_0 H(\tau)$ where C_0 is a constant, and $H(\cdot)$ is the Heaviside function, the solution is $C(t, \tau) = C_0 H(\tau - t)$. Tracer flow modelling using streamline simulation has been investigated in [*Crane and Blunt*, 1999].

5. Numerical experiments

We are working on the implementation of a streamline tracing method using a mixed-dimensional finite volume method for the pressure equation. In this paper we simulate the pressure solution of such a method using an MPFA-method.

We consider the test case in Figure 3(a). In that figure the solution domain is the rectangle BFOL, where the heavy black lines are no-flow boundaries. The fracture is the cross-hatched area in middle. A source is located in corner B and edge IL and edge FG are outflow boundaries. We have a lower, upper, and a fracture region, with permeability $\boldsymbol{K}_1 = \boldsymbol{I}$, $\boldsymbol{K}_2 = 10\boldsymbol{I}$, and $\boldsymbol{K}_3 = 1000\boldsymbol{I}$, respectively, where $\boldsymbol{I}$ is the identity matrix, see Figure 3(a). A single-phase pressure equation will be solved on a given discretization of the domain using an MPFA-method, which here, due to a Cartesian grid, reduces to TPFA. The fracture is discretized as a 2D element with a small width d. The solution will give fluxes on each grid cell edge. Some streamlines for this solution is shown in Figure 3(b).

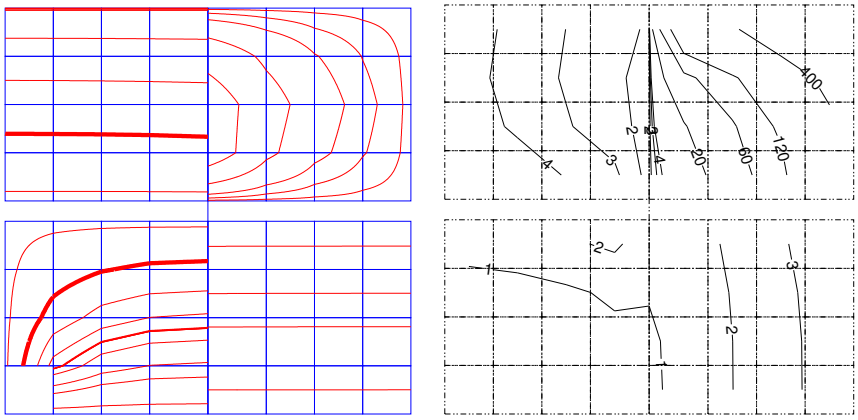
Next, the 2D fracture cells are collapsed into corresponding 1D elements, with associated fluxes taken from the 2D solution. Now we apply the streamline tracing method for fractures, see Section 3. The result is shown in Figure 3(b). The heavy streamline



(a) The fracture case.

(b) Streamlines traced for the reference solution

FIGURE 3. Generating a reference solution



(a) Streamlines traced through a fracture

(b) Tracer arrival time contours

FIGURE 4. Streamline tracing through a fracture

in Figure 4(a) is drawn to indicate where a particular streamline enters and exits the fracture.

Tracer arrival times (see Section 4) are easily computed from the traced streamlines in Figure 4(a). Recall that when tracing streamlines, we can also compute the time-of-flight (TOF) for each grid cell a streamline traverses. If we use the model in Section 4, we get from Eq. (11) that the TOF is equal to the tracer arrival time. Tracer arrival time contours for the case are shown in Figure 4(b).

6. Summary and conclusions

We have described a streamline method for handling flows on irregular grids modelling faulted and fractured media. An important application of this methodology may be risk assessment related to geological storage of CO_2. Potentially this application will

require numerous calculations of multiple realizations of geological formations described in a stochastic fashion. Traditional reservoir simulators, based on e.g. finite difference or finite volume methods, require considerable computational cost. In contrast, streamline methods and semi-analytical calculations constitute promising tools for this type of simulations. However, in the case of streamline simulations, this requires that the streamline methods are able to handle potential conduits for leakage like faults and fracture zones.

In this work we have described some preliminary steps in this direction. First, in the case of grids representing faults, special care is needed to handle sections of grid-edges (or faces) representing no-flow conditions, such that streamlines crossing the faults are represented accurately. Second, fractures are usually handled as lower dimensional objects in the computational grid. To determine exit times for streamlines entering a fracture, the fractures are virtually expanded to fully dimensional objects using the measure of the aperture of the fractures. We have illustrated these developments through a numerical experiment.

References

Aavatsmark, I. (2002), An introduction to multipoint flux approxmations for quadrilateral grids, *Comput. Geosci.*, *6*, 405–432.

Aavatsmark, I., E. Reiso, and R. Teigland (2001), Control volume discretization method for quadrilateral grids with faults and local refinements, *Comput. Geosci.*, *5*, 1–23.

Bachu, S., and D. Gunter (2004), Overview of acid-gas injection operations in western Canada, in *Rubin et al.* [2004].

Bastian, P. (1999), Numerical computation of multiphase flow in porous media, *Tech. rep.*, University of Kiel, habilitation thesis.

Batycky, R. (1997), A three-dimensional two-phase field scale streamline simulator. (http://www.streamsim.com/index.cfm?go=about.publications), Ph.D. thesis, Stanford University, Dept. of Petroleum Engineering.

Blunt, M., K. Liu, and M. Thiele (1996), A generalized streamline method to predict reservoir flow, *Petroleum Geoscience*, *2*, 259–269.

Celia, M., S. Bachu, J. Nordbotten, S. Gasda, and H. Dahle (2004), Quantitative estimation of CO_2 leakage from geological storage: analytical models, numerical models, and data needs, in *Rubin et al.* [2004].

Crane, M., and M. Blunt (1999), Streamline-based simulation of solute transport, *Water Resour. Res.*, *35*(10), 3061–3078.

Damen, K., A. Faaij, and W. Turkenburg (2004), Health, safety and environmental risks of underground CO_2 sequestration, in *Proceedings of the GHGT-7, seventh international conference on Greenhous Gas Control Technologies, Vancouver, Canada, 5-9 September.*

Gerritsen, M., K. Jessen, B. Mallison, and J. Lambers (2005), A fully adaptive streamline framework for the challenging simulation of gas-injection processes. SPE 97270, in SPE ATCE.

Hægland, H., H. Dahle, G. Eigestad, K.-A. Lie, and I. Aavatsmark (2005), Improved streamlines and time-of-flight for streamline simulation on irregular grids, submitted to *Advances in Water Resources*.

Jimenez, E., K. Sabir, A. Datta-Gupta, and M. King (2005), Spatial error and convergence in streamline simulation. SPE 92873, in *SPE Reservoir Simulation Symposium, Woodlands, Texas, 31 Jan. -2 Feb.*

King, M., and A. Datta-Gupta (1998), Streamline simulation: A current perspective, *In Situ*, *22*(1), 91–140.

Martin, V., J. Jaffré, and J. Roberts (2005), Modeling fractures and barriers as interfaces for flow in porous media, *SIAM J. Sci. Comput.*, *26*(5), 1667–1691.

Matringe, S. (2004), Accurate streamline tracing and coverage, Master's thesis, Stanford university, Dept. of petr. eng.

Matringe, S., and M. Gerritsen (2004), On accurate tracing of streamlines, in *Proceedings of the SPE Annual Technical Conference and Exhibition, Houston, Texas, 26-29 Sep.*

Matringe, S., R. Juanes, and H. Tchelepi (2005), Streamline tracing on general triangular and quadrilateral grids. SPE 96411, in SPE ATCE.

Pollock, D. (1988), Semi-analytical computation of path lines for finite-difference models, *Ground Water*, *26*(6), 743–750.

Prévost, M. (2003), Accurate coarse reservoir modeling using unstructured grids, flow-based upscaling and streamline simulation, Ph.D. thesis, University of Stanford, http://geothermal.stanford.edu/pereports/search.htm.

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

Reichenberger, V., H. Jakobs, P. Bastian, and R. Helmig (2006), A mixed-dimensional finite volume method for two-phase flow in fractured porous media, *Advances in Water Resources*. In Press.

Rubin, E., D. Keith, and C. Gilboy (Eds.) (2004), *Proceedings of 7th International Conference on Greenhouse Gas Control Technologies. Volume 1: Peer-Reviewed Papers and Plenary Presentations, IEA Greenhouse Gas Programme, Cheltenham, UK*.

Settari, A., and K. Aziz (1972), Use of irregular grid in reservoir simulation, *SPE J.*, pp. 103–114.

SPE ATCE (2005), *Proceedings of the SPE Annual Technical Conference and Exhibition, Dallas, Texas, 9-12 Okt.*

Sun, S., X. Gai, and M. Wheeler (2005), Streamline tracing on unstructured grids, in SPE ATCE.

Paper E

Comparison of cell- and vertex centered discretization methods for flow in a two-dimensional discrete fracture-matrix system

*

* Submitted to *Advances in Water Resources*, 2009

COMPARISON OF CELL- AND VERTEX CENTERED DISCRETIZATION METHODS FOR FLOW IN A TWO-DIMENSIONAL DISCRETE FRACTURE-MATRIX SYSTEM

H. HÆGLAND, A. ASSTEERAWATT, H. K. DAHLE, G. T. EIGESTAD, AND R. HELMIG

Abstract. Simulations of flow for a discrete fracture model in fractured porous rocks have gradually become more practical, as a consequence of increased computer power and improved simulation and characterization techniques. Discrete fracture models can be formulated in a lower-dimensional framework, where the fractures are modeled in a lower dimension than the matrix, or in an equi-dimensional form, where the fractures and the matrix have the same dimension.

When the velocity of the flow field is needed explicitly, as in streamline simulation of advective transport, only the equi-dimensional approach can be used directly. The velocity field for the lower-dimensional model can then be recovered by post-processing which involves expansion of the lower-dimensional fractures to equi-dimensional ones.

In this paper, we propose a technique for expanding lower-dimensional fractures and we compare two different discretization methods for the pressure equation; one vertex-centered approach which can be implemented as either a lower- or an equi-dimensional method, and a cell-centered method using the equi-dimensional formulation. The methods are compared with respect to accuracy, convergence, condition number, and computer efficiency.

1. Introduction

Fractured formations occur commonly in nature. Folding, faulting, and subsidence of sediments over geologic time cause fracturing. The more brittle the rock, the more intensely it fractures. Fractures can range in size from microns to hundreds of kilometers, and the accurate modeling of flow through such systems is important for many types of problems, including the management of water and energy resources and CO_2-sequestration [14, 15, 27].

Mathematical models based on continuum theory for quantifying flow and transport through fractured rocks can be classified into 1) continuum models, or 2) discrete fracture models, or 3) hybrid models, see e.g. [11, 14, 27, 51, 55, 60]. The models differ in their representation of the heterogeneity of the fractured media, and whether they are formulated in a deterministic or a stochastic framework.

In this paper we consider a discrete fracture model for flow in fracture-matrix system in two space dimensions (2D). Discrete fracture models [4, 23, 27, 60] allow quantification of many flow and transport phenomena that

Date: February 6, 2009.

Key words and phrases. Streamline tracing, Time-of-flight, Irregular grids.

are not adequately captured by single- or multi-continuum models. An advantage of the discrete fracture approach is that it can account explicitly for the effects of individual fractures on fluid flow and transport.

However, the contrast in permeability between fractures and the porous matrix may span many orders of magnitude and can vary highly in space [27], which is a major difficulty in modeling fluid flow in fractured rock [23]. Furthermore, due to the complex geometry of the model, unstructured grids are required for the discretization of the domain. Also, the heterogeneous and anisotropic behavior of the permeability is challenging for the numerical modeling of the system, and proper averaging of permeability between the computational cells is crucial [29].

Transport models are used for understanding and predicting the geological system using tools like break-through curves, arrival time and storage behavior. The outcome of transport simulations strongly depends on the distribution of velocities in the fractures [23]. The velocity distribution cannot, in general, be determined analytically, and must be obtained numerically from the solution of a flow (pressure) equation. Hence, desirable properties of a numerical scheme for flow simulation include: a) efficient approximation and solution of the flow (pressure) equation, b) local mass conservation, and c) high-accuracy approximation of the velocity field. The importance of efficiency is further increased by the need for considering Monte Carlo simulations.

Two approaches exist for discrete fracture models. If the matrix is (almost) impermeable and only interconnected fractures contribute to the flow, discrete fracture network models are appropriate [4, 23]. In this work we will consider a discrete-fracture-matrix (DFM) model [27]. This approach apply when both fractures and matrix play a significant role for the flow and transport processes and the model can not be homogenized. Two model approaches exist for the DFM model: In the DFM lower-dimensional formulation (DFML) [41, 44, 45, 49, 58], fractures are modeled as line segments in 2D, and as planar regions in 3D, whereas in the DFM equi-dimensional formulation (DFME) [13, 19, 46, 57, 58, 66], fractures are modeled in the same dimension as the matrix. The DFML has been more common than the DFME, since modeling fractures as lower-dimensional objects simplifies grid generation and data requirements. However, when the velocity of the flow field is needed explicitly, as in streamline simulation of advective transport [24], only the equi-dimensional approach can be used directly. The velocity field for the lower-dimensional model must be recovered by a post-processing step which involves an expansion of the lower-dimensional fractures to equi-dimensional ones [38].

In this paper, we consider a discrete fracture model where fractures are modeled as lower dimensional objects [58] in a geostatistical fracture generator [8]. The domain is assumed to be in 2D, such that fractures are modeled as line segments. A technique for expanding lower-dimensional fractures is proposed, and we investigate different discretization methods for flow on both DFML grids and DFME grids. Extensions to 3D is briefly discussed

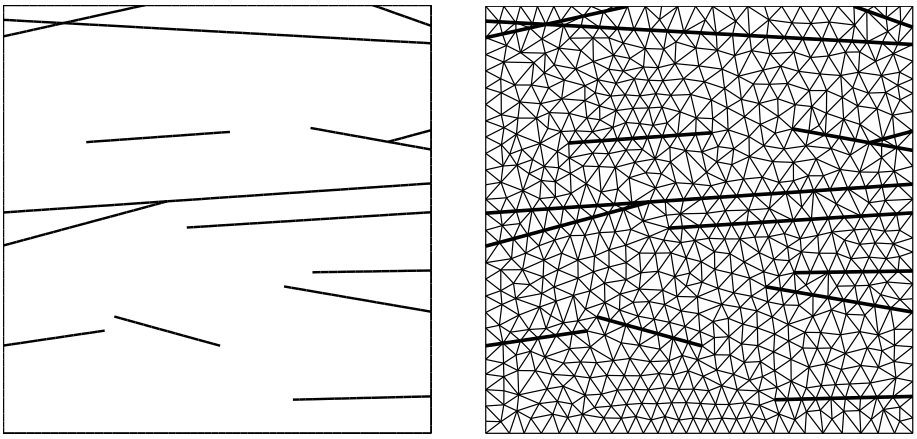


FIGURE 1. Left: A realization from the geostatistical fracture generator FRAC3D [62]. The fractures in black are modeled as lower dimensional objects, i.e. line segments. Right: Discretization of the realization in the left figure using the mesh generator ART [34]. Note that some estimate of fracture width (or aperture) must be provided by the fracture generator.

in the next section. A vertex-centered box scheme is compared with a cell-centered multi-point flux approximation. Section 3 introduces the vertex-centered and cell-centered finite volume methods. Section 4 compares the methods numerically for several test cases of increasing complexity. To illustrate that the fracture expansion also facilitate tracing of streamlines, we provide streamline calculations which demonstrate the convergence of the numerical velocity field. Finally, Section 5 concludes the paper.

2. Expansion of fractures

We consider fractures modeled as lower dimensional objects [58] by a geostatistical fracture generator [8]. A sample 2D realization is shown in Figure 1 (left). When fractures in a lower-dimensional model are subsequently expanded to have the same dimension as the matrix, we obtain a so-called equi-dimensional fracture model.

The fractures may be expanded before, or after the discretization of the fracture field. Usually the fractures will be very thin compared to the size of the domain. If a relative coarse mesh is being used around the fractures, e.g., as in Figure 1 (right), the size of the mesh elements will be much larger than the fracture aperture. In this case the fracture must be expanded *after* the grid generation. For a very fine mesh, however, the size of the triangles may be similar to the fracture aperture, or smaller. In this case, the fractures should be expanded *before* the grid is generated.

We will henceforth only consider cases where the fractures are expanded after the triangulation. We also assume that a constant aperture can be associated with each fracture segment, using a parallel plate model for the

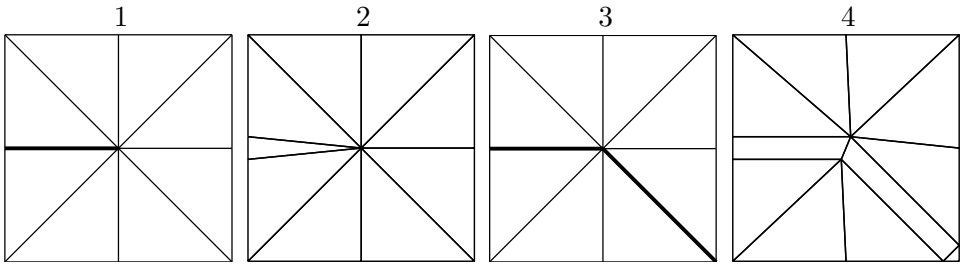


FIGURE 2. The cases $\varrho = 1$ and $\varrho = 2$. Sub figure 1) and 2): A single fracture ending. Sub figure 3) and 4): Two fractures meeting

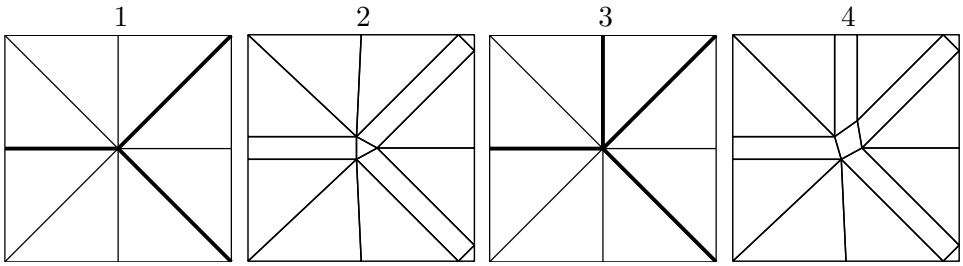


FIGURE 3. The cases $\varrho = 3$ and $\varrho = 4$. Sub figure 1) and 2): Three fractures meeting. Sub figure 3) and 4): Four fractures meeting

fractures [12]. The fractured domain is in 2D and will be discretized with triangles, such that the fractures coincides with the triangle edges. To simplify the presentation, we will assume that all fractures have the same aperture.

2.1. **Interior points.** The expansion algorithm is based on the number ϱ of fractures meeting at a vertex. We consider the cases of ϱ equal to 1, 2, 3, 4, and 5. Higher degrees are treated similar to degree 5. The reason for considering case 3, 4 and 5 separately, is related to the type of acceptable elements in the grid: triangles and quadrilaterals are permitted and can be used for the central element when $\varrho = 3$ and $\varrho = 4$, respectively. For $\varrho \geq 5$, the central element must be split into sub triangles, as shown later.

The case $\varrho = 1$, corresponds to a single fracture ending at an interior vertex, see Figure 2. A single fracture can be expanded in variety of ways, see e.g. [35, 53, 52, 54, 59]. This topic will not be investigated further in this work, here we use a single triangle as shown in Figure 2 (Left). The case $\varrho = 2$ is handled using two trapezoidal elements, see Figure 2 (Right). The case $\varrho = 3$, is shown in Figure 3 (Left). If the permeability of the three fractures is the same, this case can be handled using three trapezoids and a central triangle. If the permeability of the three fractures is different, the permeability of the central triangle can be calculated using a generalized harmonic mean [45], or the central triangle can be split into three sub triangles, each of which correspond to one of the fracture edges. The case $\varrho = 4$ is shown in Figure 3 (Right). In this case four trapezoids and a central quadrilateral are used, if the permeability of the fractures is different, the central quadrilateral may be split into four sub triangles. The

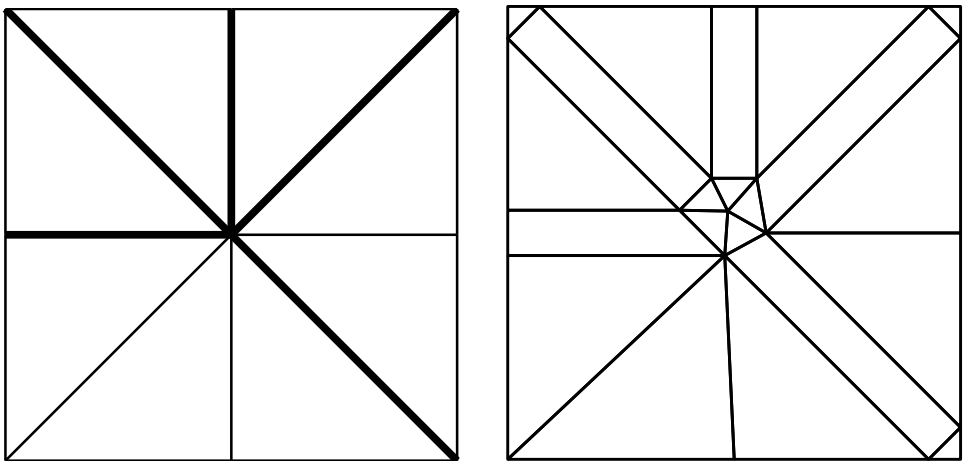


FIGURE 4. Five fractures meeting

case $\varrho = 5$ is shown in Figure 4. Cases of $\varrho > 5$ are handled in a similar fashion.

2.2. **Boundary points.** Boundary vertices with ending fractures are handled similar to interior vertices. If the boundary is flat at the vertex and only a single fracture is ending at the vertex, see e.g. the vertical fracture in Figure 4 (Right), then no extra element at the boundary is needed. If the boundary is not flat, or two or more fractures are meeting at the vertex, extra elements, either triangles or quadrilaterals are inserted at the boundary vertex, see also Figure 4 (Right).

2.3. **Extension to 3D.** Obviously, the expansion procedure in 3D is more complicated than for the 2D case. In the lower-dimensional form, each fracture will be modelled as a plane. A single plane not crossing any other planes, can then be expanded with hexahedral elements. Two planes will cross each other along a line if they are not parallel. The intersection line can also be expanded with hexahderal elements. Three non-parallel planes (with normal vectors not lying in the same plane) cross each other along three lines and in a single point, see [68]. The crossing point would be discretized with a hexahedral element. Four planes will cross each other along six lines and in three points. The points corresponds to intersections of the three possible groups of three planes taken from a group of four planes. It is highly unlikely that four planes cross in a single point. More than 4 planes will cross along lines and in points corresponding to groups of 3 planes.

3. Finite volume methods for elliptic problems

The flow equation for a single phase flowing in a porous medium is given by

$$-\nabla \cdot \boldsymbol{K} \nabla P = f, \tag{1}$$

for a permeability tensor $\boldsymbol{K}$, pressure P, and a source term f.

For general anisotropic and inhomogeneous medium, the permeability tensor is represented by a space-dependent full tensor. Layered and fractured

media involving large discontinuities in the permeability require numerical schemes with a continuous flux and a proper definition of transmissibility across cell-edges. Furthermore, the discrete features of the media and the general geometry of the media require the use of general unstructured grids. Since a simple two-point flux approximation is only applicable for the special case of a $\boldsymbol{K}$-orthogonal grid [2], more robust methods must be considered in our context.

Many schemes that preserve flux continuity on general unstructured grids have been developed, like mixed-finite element methods [16, 7, 6, 5], discontinuous Galerkin methods [20], mimetic finite difference methods [43, 61] and control-volume mixed finite element methods [17].

In this paper, we focus on two different finite volume methods (FVMs) [33] applicable to the discrete fracture model. The idea of a FVM is to discretize the domain into a mesh of finite volumes (or control volumes). Then the divergence form of the equation is exploited by integrating over a finite volume and using Gauss' theorem to convert the result into surface integrals which are discretized [50].

The most common FVM schemes are either cell-centered (block-centered), in which the primal grid cells are used to define the control volumes, or vertex-centered (point-distributed), in which control volumes are constructed around the primal grid vertices.

3.1. **The box method.** The box methods [48] are vertex-centered FVMs that can be formulated as a finite element method. Generally, a finite element method involves two spaces, the test- and the trial-space. The box method can be cast in the finite element setting with a trial space of piecewise polynomials over the primary grid, and a test space over the dual grid [48]. Since the trial functions will be different from the test functions, the box scheme can be treated as a Petrov-Galerkin finite element method [21].

For the box-scheme considered herein, the test functions are piecewise constant, and the trial functions are conforming piecewise linear polynomials. This kind of box scheme has also been called control-volume finite element method [9], finite volume element method [18], generalized difference method [48], and subdomain collocation method [42, 69].

Consider first the case of no fractures. The domain is discretized with triangular and/or quadrilateral elements E_j, $j = 1, \ldots, N^e$, and vertices $\boldsymbol{\mathcal{V}}_i$, $i = 1, \ldots, N^v$, where N_e is the total number of elements in the grids, and N_v is the total number of vertices in the grid. In the box method, one algebraic equation corresponds to every vertex in the primary mesh. The boxes are usually constructed as a dual mesh of an underlying grid. There are various ways to introduce the dual mesh, e.g., the Voronoi mesh based on the circumcenter [18, 48], or the Donald-mesh, based on the barycenter [10, 21, 32].

For the Donal-mesh the secondary dual mesh is defined by connecting the barycenter (centroid) of each element E_j with edge midpoints. The dual mesh divides each element into three or four sub quadrilaterals depending on the type of element (triangle/quadrilateral). The union of all sub quadrilaterals adjacent adjacent to a vertex $\boldsymbol{\mathcal{V}}_i$ is denoted Ω_i^B, or the control volume associated with $\boldsymbol{\mathcal{V}}_i$, see Figure 5 (left).

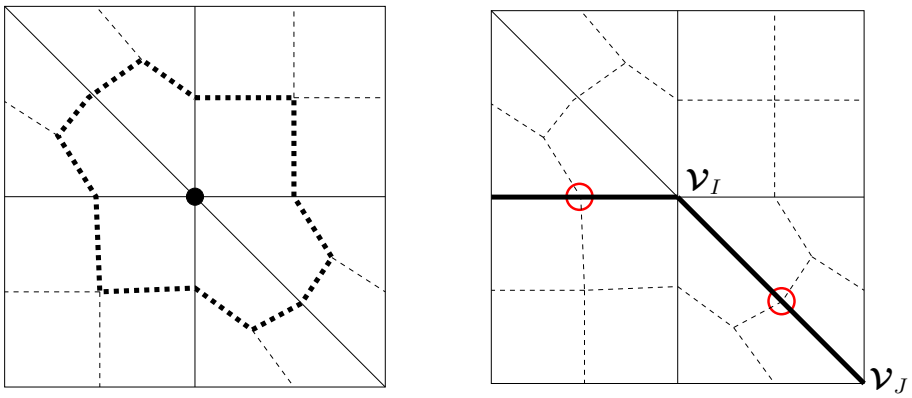


FIGURE 5. Left: Sample grid with no fractures consisting of a primary grid of two quadrilaterals and four triangles. The dual grid is indicated by dashed lines. The control volume associated with the central vertex (black circle) is indicated by the heavy dashed line. Right: Grid with fractures; intersection points indicated with red circles. A fracture intersects the dual mesh at its midpoint. The vertices $\boldsymbol{\mathcal{V}}_I$ and $\boldsymbol{\mathcal{V}}_J$ are the end points of a fracture.

As for the finite element method, the unknown in Equation (1), $P = P(x, y)$ is approximated using basis functions ϕ_i, $i = 1, \ldots, N^v$, weighted by discrete values $\hat{P}_i \equiv P(\boldsymbol{\mathcal{V}}_i)$, i.e., $P \approx \tilde{P} = \sum \phi_i \hat{P}_i$. A basis function is a piecewise continuous function such that its restriction to a triangular element will be a linear function, whereas its restriction to a quadrilateral is an isoparametric mapping of a bilinear function on a reference element [22, 64]. As usual, the basis functions ϕ_i are local functions, meaning that they are identically zero outside Ω_i^B.

The approximation of P with $\tilde{P}$ now leads to a residual in Equation (1), $\nabla \cdot \tilde{\boldsymbol{q}} - \nabla \cdot \boldsymbol{q} = \nabla \cdot \tilde{\boldsymbol{q}} - f$, where $\tilde{\boldsymbol{q}} = -\boldsymbol{K} \nabla \tilde{P}$. In the method of weighted residual [69], the weighted integral of the residual over the whole domain is set to zero,

$$\int_\Omega W_i (\nabla \cdot \tilde{\boldsymbol{q}} - f) dV = 0, \qquad i = 1, \ldots, N^v, \tag{2}$$

for each weight function W_i, $i = 1, \ldots, N^v$. The Box-method uses the simple form,

$$W_i(x) = \begin{cases} 1 & x \in \Omega_i^B \\ 0 & x \notin \Omega_i^B. \end{cases} \tag{3}$$

The application of the divergence theorem to (2) now leads to a set of N^v surface integrals over the boundaries of each control volume Ω_i^B. For triangular elements, these integrals can be evaluated analytically, whereas for quadrilateral elements the gradient of the pressure is not constant on each element, and numerical integration is required. In the numerical experiments in this paper we used a midpoint rule to evaluate the fluxes for quadrilateral elements.

Assuming an element-wise constant permeability, the N_v integrals can be expressed as

$$\sum_{j=1}^{N_{t,i}} \sum_{k=1}^{3} \gamma_{i,j,k} \hat{P}_{i,j,k} = f_i \qquad i = 1, \ldots, N^v. \tag{4}$$

Here, we have assumed a grid of only triangular elements, and the sum on k is running over the three vertices of each triangle j contributing to the control volume Ω_i^B. There are $N_{t,i}$ triangles contributing to Ω_i^B, and $\hat{P}_{i,j,k}$ refers to the pressure at local corner number k of triangle j. Furthermore, f_i is the integral of the source term f over Ω_i^B, and $\gamma_{i,j,k}$ can be expressed as

$$\gamma_{i,j,k} = \left[\frac{\boldsymbol{K}_{i,j}}{2T_{i,j}} \boldsymbol{\nu}_{i,j,k} \right] \cdot (\boldsymbol{n}_{i,j}^1 + \boldsymbol{n}_{i,j}^2). \tag{5}$$

Here $\boldsymbol{n}_{i,j}^1$ and $\boldsymbol{n}_{i,j}^2$ are the two outward normal vectors (having length equal to the length of the interface) associated with the part of $\partial\Omega_i^B$ lying within triangle j, $\boldsymbol{K}_{i,j}$ and $T_{i,j}$ are the permeability tensor and the area of triangle j, respectively. The vectors $\boldsymbol{\nu}_{i,j,k}$ and the area $T_{i,j}$ results from taking the gradient of the linear pressure variation in triangle j as shown in, e.g., [3].

The last part of the box method consists of the assembly of the coefficient matrix and the right hand side of the linear system for the discrete (vertex) pressures giving N^v linear equations for the N^v unknown vertex pressures $\hat{P}_i$. Note that, for Dirichlét boundary conditions, the number of Dirichlét boundary nodes must be subtracted from N^v, see [37] for a discussion of boundary conditions.

The linear system can be written in matrix form as

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{A}$ is the $N^v \times N^v$ coefficient matrix, $\boldsymbol{u}$ is an $N^v \times 1$ vector of the unknown vertex pressures, and $\boldsymbol{b}$ is an $N^v \times 1$ vector representing the source terms or boundary conditions. Since the linear system is symmetric positive definite, it can be solved rapidly using, e.g., the preconditioned conjugate gradient method.

3.2. **Discrete fractures.** Fractures are approximated using the parallel plate model [12]. In n-dimensional space, fractures are often modeled as $(n-1)$-dimensional objects [27, 58]. In 2D, this means that fractures are associated with the edges of the finite element mesh. Hence, edges are either matrix edges (edges containing no fractures), or fracture edges (edges containing a fracture), see Figure 5 (right). The fracture edges are denoted by $\mathcal{F}_k$, $k = 1, \ldots, N_f$, where N_f is the number of fracture edges. A fracture edge intersects a control volume boundary of the dual mesh at its midpoint, see Figure 5 (right).

Each fracture edge $\mathcal{F}_k$ will also have an associated aperture a_k, permeability k_k, and length l_k. The length l_k is the length of the fracture edge, the permeability k_k is assumed to be isotropic such that a two-point flux can be applied inside the fracture and the aperture a_k is not resolved explicitly in the grid, but used for computing fracture fluxes, as discussed next.

Consider a fracture edge $\mathcal{F}_k$ with end points at nodes $\boldsymbol{\mathcal{V}}_I$, $1 \leq I \leq N^v$, and $\boldsymbol{\mathcal{V}}_J$, $1 \leq J \leq N^v$, as in Figure 5 (right). Then the flux $Q_{i,j,k}$ out of Ω_I^B at the midpoint of $\mathcal{F}_k$ is approximated as a two-point flux

$$Q_k = \frac{k_k a_k}{l_k}(\hat{P}_I - \hat{P}_J). \tag{7}$$

The above fluxes are then taken into account when evaluating the surface integrals resulting from Equation (2), and when assembling the linear system (6).

The lower-dimensional treatment of fractures in the box methods has been popular since it allows for easy mesh generation and a well-conditioned linear systems. However, streamlines cannot be traced directly for this approach, since there is no transversal flow information in the lower-dimensional fractures. It is therefore of interest to consider equi-dimensional methods. In the next section we consider the MPFA method for equi-dimensional fractures. Note that the MPFA method also has been used with lower-dimensional fractures [40], however this approach can only be used for low-permeable fractures.

3.3. **The MPFA O-method.** In cell-centered finite volume methods the control-volumes are associated with the primary grid. This is convenient compared to vertex-centered methods, when considering discontinuous media properties combined with a quadrilateral or triangular primary mesh. In this case it is easy to align the grid edges, and hence, the control-volume boundaries with media discontinuities.

Here we use a multi-point flux O-method (MPFA O-method), briefly described below. For an overview of different MPFA schemes see [2, 1, 3, 28, 67], and references therein. The letter "O" comes from the shape of the polylines connecting the involved grid points in a cell-stensil [2]. The MPFA O-method is based on full flux continuity across cell edges, continuity of the pressure at the midpoint of the cell edges, and mass conservation for each cell. The pressures at cell edges is eliminated locally using the flux continuity constraints, and the global system is expressed in terms of the cell center pressures.

For the MPFA O-method the control volumes are associated with the original mesh, whereas for the box method, control volumes were associated with the dual mesh. By integrating Equation (1) over a control volume Ω_i^M and applying Gauss' theorem we have

$$-\int_{\partial\Omega_i^M} \boldsymbol{K}\nabla P \cdot \boldsymbol{n} dS = \int_{\Omega_i^M} f dV, \qquad i = 1, \ldots, N^e, \tag{8}$$

where $\boldsymbol{n}$ is the unit outward normal vector to $\partial\Omega_i^M$. The only unknown in the last equation is the pressure, which will be approximated at the center of each cell (cell-centered method), in contrast to the Box-method, where the pressure is approximated at the vertices of the grid (vertex-centered method). Hence, Equation (8) gives us N^e equations to determine the N^e unknown cell center pressures. By assuming a constant permeability $\boldsymbol{K}_i$ for each control volume, we can write the integral on the left hand side of

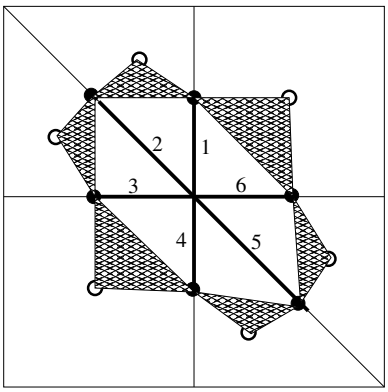


FIGURE 6. An interaction region, variational triangles, half-edges and pressure nodes, respectively, for the central vertex of a simple grid. Variational triangles are cross hatched; the half-edges are the heavy line segments numbered 1-6; solid circles indicate auxiliary (temporary) pressure nodes, and non-solid circles indicate cell center pressures.

Equation (8) as

$$\int_{\partial\Omega_i^M} -\boldsymbol{K}_i \nabla p \cdot \boldsymbol{n} dS = \sum_{j=1}^{N_{e,i}} \int_{\Gamma_{i,j}} -\boldsymbol{K}_i \nabla p \cdot \boldsymbol{n}_{i,j} dS = \sum_{j=1}^{N_{e,i}} Q_{i,j}, \; i = 1, \ldots, N^e, \tag{9}$$

where $N_{e,i}$ is the number of boundary edges of Ω_i^M, i.e., $N_{e,i}$=3 for a triangle and $N_{e,i}$=4 for a quadrilateral, $\Gamma_{i,j}$ is boundary edge number j of Ω_i^M, $\boldsymbol{n}_{i,j}$ is the corresponding outward unit normal vector, and $Q_{i,j}$ is defined by Equation (9). In order to determine the fluxes $Q_{i,j}$ in Equation (9) in terms of the cell center pressures, we use a multi-point flux approximation (MPFA).

The MPFA method is based on dividing the grid into so-called interaction regions $\mathcal{I}_i, i = 1, \ldots, N^v$, associated with each vertex $\boldsymbol{\mathcal{V}}_i$ of the grid. The interaction regions are equal to the control-volumes for the box method, see the previous section. For each interaction region $\mathcal{I}_i$ there is a set of half-edges $E_{i,j}$, see Figure 6. On each side of a half-edge there will be two grid cells, denoted "L" and "R". Furthermore, each interaction region also contains a set of variational triangles $\mathcal{T}_{i,j}^{\alpha}$, $\alpha = L, R$, one for each cell α of half-edge $E_{i,j}$, see Figure 6.

Within each $\mathcal{T}_{i,j}^{\alpha}$, a linear pressure variation is assumed. This is done by fixing the pressure at the corners of each $\mathcal{T}_{i,j}^{\alpha}$. Thus, in addition to the cell center pressures, we introduce (temporary) pressures at the points indicated with the solid circles in Figure 6. The linear pressure in each $\mathcal{T}_{i,j}^{\alpha}$ is now assumed to be valid also for calculating the flow across each of the two half-edges within $\mathcal{I}_i$ that are closest to $\mathcal{T}_{i,j}^{\alpha}$, i.e., those with a common point with $\mathcal{T}_{i,j}^{\alpha}$.

Hence, the pressure gradient will be constant on each side of a half-edge. From Equation (8) we see that the flow rate $\boldsymbol{q}$ must be given as

$$\boldsymbol{q} = -\boldsymbol{K}\nabla P. \tag{10}$$

Substituting the constant pressure gradient into the last equation, we can express the flux across each half-edge $E_{i,j}$ as

$$q^{\alpha}_{i,j} = \sum_{k=1}^{3} t^{\alpha}_{i,j,k} P(\boldsymbol{x}^{\alpha}_{i,j,k}), \tag{11}$$

where

$$t^{\alpha}_{i,j,k} = \frac{|E_{i,j}|}{2T^{\alpha}_{i,j}} \left(\boldsymbol{K}^{\alpha}_{i,j}\boldsymbol{\nu}^{\alpha}_{i,j,k}\right) \cdot \boldsymbol{n}_{i,j}, \tag{12}$$

and $P(\boldsymbol{x}^{\alpha}_{i,j,k})$ is the pressure at corner k of $\mathcal{T}^{\alpha}_{i,j}$. The sign of the unit normal vector $\boldsymbol{n}_{i,j}$ to $E_{i,j}$ is not important, but the same sign must be chosen for each value of α. In the last equation, $|E_{i,j}|$ is the length of $E_{i,j}$; $T^{\alpha}_{i,j}$ and $\boldsymbol{\nu}^{\alpha}_{i,j,k}$ are the area and normal vectors of $\mathcal{T}^{\alpha}_{i,j}$, resulting from taking the gradient of a the linear pressure, see, e.g., [3].

Flux continuity across edge $E_{i,j}$ can now be expressed as $q^{L}_{i,j} = q^{R}_{i,j}$, or

$$A_i\boldsymbol{u}_i + B_i\boldsymbol{v}_i = C_i\boldsymbol{u}_i + D_i\boldsymbol{v}_i, \qquad i = 1, \ldots, N^e, \tag{13}$$

where $\boldsymbol{u}_i$ is a vector of cell center pressures, and $\boldsymbol{v}_i$ contains the auxiliary pressures (which we want to eliminate), and the matrices A_i, B_i, C_i, and D_i, contains the t-coefficients. Note that since the pressure is required to be continuous at the auxiliary pressure nodes, i.e., at the solid circles in Figure 6, there will be only one unknown at these points, and hence each $\boldsymbol{v}_i$ has exactly N^E_i components, where N^E_i is the number of half-edges within $\mathcal{I}_i$. It can also be shown that the matrices B_i and D_i are square $N^E_i \times N^E_i$ matrices.

We here assume that the interaction region $\mathcal{I}_i$ correspond to an interior corner; for boundary corners and implementation of boundary conditions, see [37, 31]. For interior corners, the vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ will also contain the same number (N^E_i) of unknowns, such that A_i and C_i are also square $N^E_i \times N^E_i$ matrices.

Due to the continuity condition in Equation (13), the flux $q_{i,j}$ across edge $E_{i,j}$ is well defined as either $q^{L}_{i,j}$ or $q^{R}_{i,j}$, and can be expressed as, e.g., $q_{i,j} = q^{L}_{i,j}$, which is expressed in matrix form as

$$\boldsymbol{q}_i = A_i\boldsymbol{u}_i + B_i\boldsymbol{v}_i, \tag{14}$$

where the j-th component $\boldsymbol{q}_i$ is $q_{i,j}$. Finally, we eliminate $\boldsymbol{v}_i$ in Equation (14) by using Equation (13),

$$\boldsymbol{q}_i = (A_i + B_i(B_i - D_i)^{-1}(C_i - A_i))\boldsymbol{u}_i = T_i\boldsymbol{u}_i \tag{15}$$

Since the components of $\boldsymbol{q}_i$ represents fluxes over half edges adjacent to a given corner $\boldsymbol{\mathcal{V}}_i$, the flux over edge $\Gamma_{k,j}$ of grid cell Ω^M_k (see Equation (9)) can be found be combining certain components $\boldsymbol{q}_i$, $i = A, B$, where $\boldsymbol{\mathcal{V}}_A$ and $\boldsymbol{\mathcal{V}}_B$ are the end points of $\Gamma_{k,j}$.

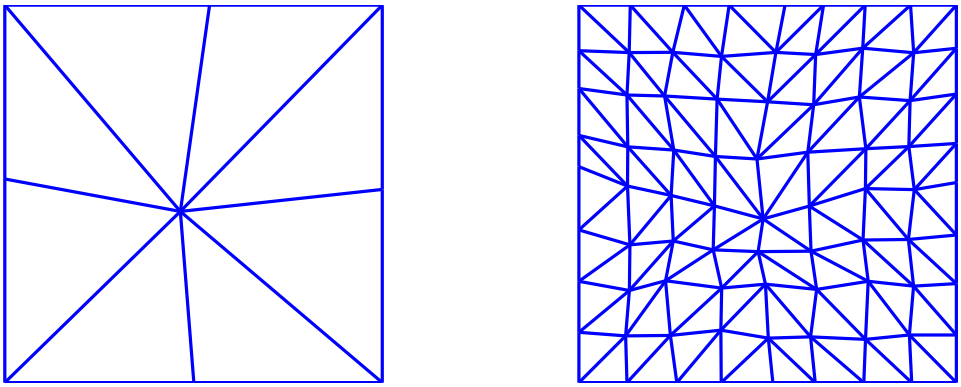


FIGURE 7. Left: Initial grid. Right: Level 2 refined grid.

In this way, we assemble expressions for each of the $Q_{i,j}$ in Equation (9) and inserting these into (8), we obtain N^e linear equations that can be solved for the N^e unknown cell center pressures. The linear system is expressed as

$$A\boldsymbol{u} = \boldsymbol{b}, \tag{16}$$

where A is the $N^e \times N^e$ coefficient matrix, $\boldsymbol{u}$ is an $N^e \times 1$ vector of the unknown cell center pressures, and $\boldsymbol{b}$ is an $N^e \times 1$ vector representing the source terms for each Ω_k^M, $k = 1, \ldots, N^e$, see Equation (8).

4. Numerical experiments

In this section we solve Equation (1) numerically, using the box method and the MPFA method.

4.1. **Homogeneous case.**

We first consider the homogeneous case with no fractures. By investigating this simple case first, we will get a feeling of how the methods perform, including the sensitivity of the solution to different boundary conditions and to grid perturbation, before extending to the more complex cases including fractures.

Assume $\boldsymbol{K} \equiv 1$, no source terms ($f \equiv 0$), and Dirichlet boundary conditions in a domain $\Omega = [0, 2] \times [-1, 1]$. The boundary conditions were imposed by one of three different analytical solutions,

$$P_1(x, y) = x^2 - y^2 + xy + x + y + 1, \tag{17}$$

$$P_2(x, y) = x^3 - 3xy^2 + 3x^2y - y^3 + x^2 - y^2 + xy + x + y + 1, \tag{18}$$

$$P_3(x, y) = \cos(x)\cosh(y). \tag{19}$$

The influence of the following boundary conditions were investigated:

- Type 1: Dirichlét boundary conditions on the whole boundary,
- Type 2: Neumann boundary conditions on the whole boundary, and the pressure fixed at an interior point, and
- Type 3: Mixed boundary conditions, with Neumann boundary conditions at the top and bottom, and Dirichlét at the left and right boundary.

The convergence tests were performed on a sequence random grids, starting with the coarsest grid, as shown in Figure 7 (left). Five refinement levels was

TABLE 1. Convergence rates for pressure and flux in terms of DOFs.

	MPFA1	MPFA2	MPFA3	BOX1	BOX2	BOX3
	Pressure					
P_1	-.96	-1	-.99	-.91	-1	-.93
P_2	-.97	-1	-.99	-.91	-1	-.97
P_3	-.97	-.94	-.99	-.95	-1	-.97
	Flux					
P_1	-.57	-.52	-.52	-.46	-.49	-.47
P_2	-.57	-.5	-.5	-.47	-.49	-.48
P_3	-.58	-.49	-.5	-.47	-.49	-.48

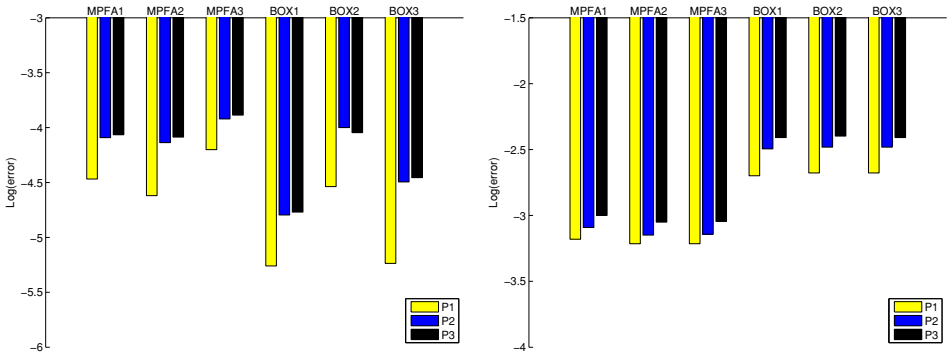


FIGURE 8. Accuracy at 3950 DOFs obtained by interpolating between the two closest grids. Left: Pressure. Right: Flux.

considered; each refinement was obtained by splitting each triangle of the original grid into four sub triangles, leading to a refined grid with four times as many cells as the original. Then each grid vertex was given a random perturbation. The level 2 refined grid is shown in Figure 7 (right). The smallest angle of any element were 36, 32, 24, 18, 14, and 9 degrees, for the coarsest to the finest grid.

The convergence rates for pressure and flux based on the discrete L^2 error [31] are shown in Table 1. The number after the method name refers to the boundary condition, e.g., MPFA1 is the MPFA method using boundary condition of type-1. Note that the MPFA method has more degrees of freedom (DOFs) than the box method for the same grid, since we use a triangular grid and the MPFA method is a cell-cent red method, whereas the box method is vertex-centered. We therefore display convergence results as a function of the DOFs; the convergence rate in terms of the mesh size parameter h can be obtained approximately by a multiplication by -2.

The results in Table 1 are in agreement with the theoretical results presented in the literature for the box method [32] and for the MPFA method [47]. The results show a second order convergence in pressure and first order in flux.

The accuracy at 3950 DOFs are shown in Figure 8. The results show that

TABLE 2. Runtime in seconds per degree of freedom (R.T./DOF) for each method as a function of grid refinement level.

		$\mathcal{G}_0$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_4$	$\mathcal{G}_5$	$\mathcal{G}_6$
R.T./DOF	BOX	0.04	0.0044	0.0027	0.0024	0.0024	0.0025	0.0032
	MPFA	0.0025	0.0031	0.0027	0.0027	0.0028	0.0038	0.014

the MPFA method is most accurate for flux, and the BOX method is most accurate for the pressure.

As noted in Section 3.1, the system matrix for the box method is symmetric and positive definite, and we used a Cholesky factorization to solve the corresponding linear system. An other alternative would be the preconditioned conjugate gradient method, see [25] for a comparison. The system matrix for the MPFA method is, however, not symmetric [30], and the UMFPACK [26] LU-factorization is used for this system. Table 2 shows runtimes for the methods obtained for the Dirichlét boundary condition. The runtimes are divided by the degrees of freedom to provide a fair comparison between the methods. Note also that an extra refinement level, $\mathcal{G}_6$, is included in the table. The number of elements for $\mathcal{G}_i$, $i = 0, \ldots, 6$ was 8×4^i, respectively. Hence $\mathcal{G}_6$ contained 32768 elements. The table shows that the runtime per degree of freedom is almost constant for the box method for refinement levels 3, 4, and 5, and then starts to increase slightly for the last refinement level. For the MPFA method, the runtime is almost constant for first 4 refinement levels, and then starts to increase. For the last refinement ($\mathcal{G}_6$), the box method is approximately 4 times faster than the MPFA method.

The numerical test in this section indicates that:

- For fine grids (more than 20000 DOFs), the box method will be faster the MPFA method;
- The box method is most accurate for the pressure, whereas the MPFA method is most accurate for the flux;
- The convergence rate of the MPFA method and the box method were similar for both flux and pressure.

4.2. **Single fracture.** In this section we solve Equation (1) for the case where the permeability tensor $\boldsymbol{K}$ is assumed to be the piecewise constant scalar,

$$\boldsymbol{K}(x,y) = \begin{cases} 1, & (x,y) \in \Omega_1 \\ k_F, & (x,y) \in \Omega_2 \\ 1, & (x,y) \in \Omega_3. \end{cases} \tag{20}$$

The solution domain $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ is shown in Figure 9 (left). The domain Ω_2 will act as a single horizontal fracture with aperture ϵ.

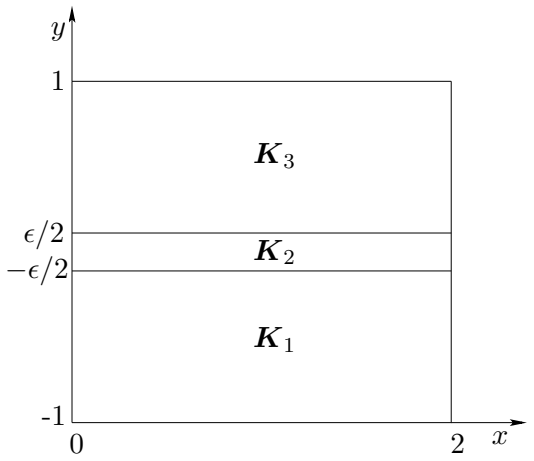


FIGURE 9. A domain with a single horizontal fracture.

We will consider the following analytical solution,

(21)
$$P_F(x,y) = \begin{cases} k_F \cos(x)\cosh(y) + (1-k_F)\cosh(\epsilon)\cos(x) & (x,y) \in \Omega_1 \\ \cos(x)\cosh(y) & (x,y) \in \Omega_2 \\ k_F \cos(x)\cosh(y) + (1-k_F)\cosh(\epsilon)\cos(x) & (x,y) \in \Omega_3, \end{cases}$$

where the source term in Equation (1) is given by

(22)
$$f(x,y) = \begin{cases} (1-k_F)\cosh(\epsilon)\cos(x), & (x,y) \in \Omega_1 \\ 0, & (x,y) \in \Omega_2 \\ (1-k_F)\cosh(\epsilon)\cos(x), & (x,y) \in \Omega_3. \end{cases}$$

The boundary conditions will be determined by the analytical solution, and we consider the type-1, type-2, and type-3 boundary conditions as described in the previous section.

The numerical solution is computed in five different ways. First, we represent the fracture in Figure 9 using a lower dimensional object, i.e., a horizontal line segment at y=0. This method will be denoted the BOXL ("L" for lower-dimensional) method. The other four methods are based on the equi-dimensional representation of the fracture. As discussed in Section 2, the equi-dimensional fractures are obtained by expanding the fracture *after* the domain has been triangulated. Two alternatives are investigated. The first uses quadrilateral elements in the fracture. This is shown in Figure 10 (left) for the coarsest grid and for an exaggerated fracture aperture. Using this grid, we solve the pressure equation using both the MPFA method and the box method, leading to the methods BOXQ and MPFAQ ("Q" for quadrilateral). The last alternative uses triangular elements in the fracture, shown in Figure 10 (right), leading to the methods BOXT and MPFAT.

However, as it turned out that the errors for the methods using triangular elements in the fracture could not be distinguished visually from the errors of the same method using quadrilateral elements in the plots presented below. Hence, to simplify the presentation, we do not display any results for BOXT and MPFAT methods in the following.

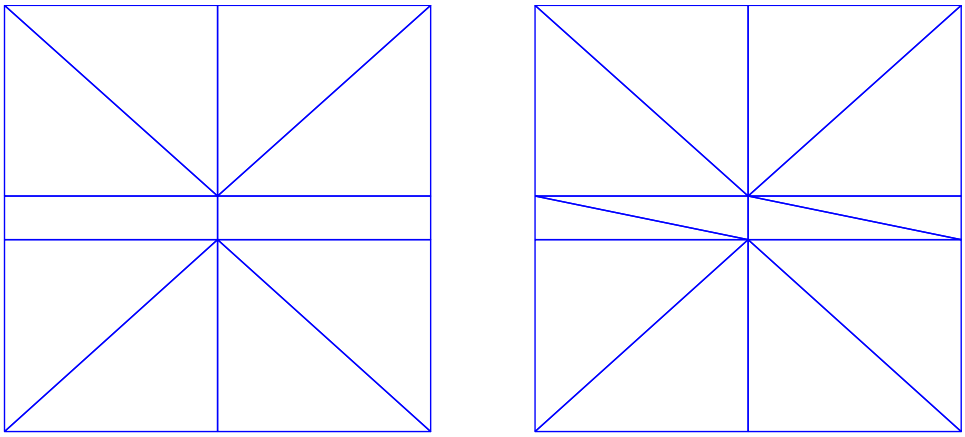


FIGURE 10. Different discretizations of an expanded fracture. The aperture of the horizontal fracture at the center of the domain is exaggerated for illustration purposes. Left: Quadrilateral elements in the fracture. Right: Triangular elements.

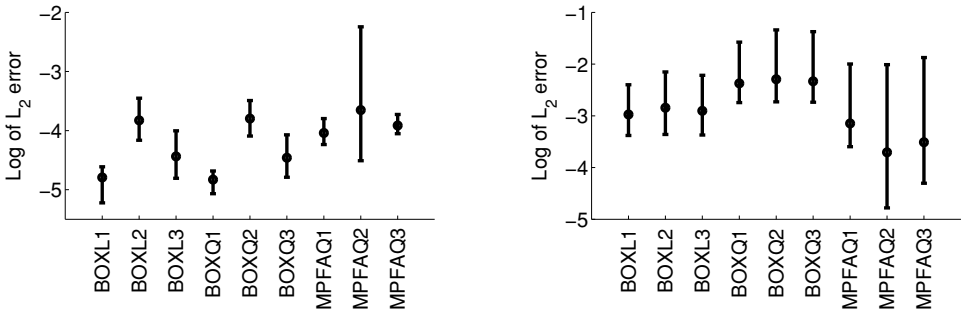


FIGURE 11. Errorbar plot for the L^2 error in pressure at 3950 DOFs. Left: Pressure. Right: Flux

We would like to investigate both the influence of the thickness of fracture, and the influence of the permeability contrast between the fracture and the matrix on the system behavior. However, in a later section we show that these two factors are not independent in their influence on the condition number of the system. We have therefore chosen to investigate only the permeability contrast here. Then, in order to include a range of both low-permeable fractures and high-permeable fractures we consider the following cases for the fracture permeability:

$$k_{F,i} = 10^{i-6}, \qquad i = 1, 2, \ldots, 11, \tag{23}$$

and we chose to fix the fracture fracture aperture to

$$\epsilon = \text{1e-4} \tag{24}$$

for all cases.

The results for accuracy are shown in Figure 11. As in the previous section, we evaluate the accuracy at 3950 DOFs, and the last figure in the

TABLE 3. Convergence rates for pressure in terms of DOFs. Note that the rates for the BOXQ1, BOXQ3, MPFAQ1, and MPFAQ3 are not included in the table below since they were -1 for all cases of the fracture permeability.

Perm.	BOXL1	BOXL2	BOXL3	BOXQ2	MPFAQ2
1e-05	-0.99	-1.1	-1	-0.9	-0.5
1e-04	-0.99	-1.1	-1	-0.92	-0.49
1e-03	-0.99	-1.1	-1	-0.92	-0.53
1e-02	-0.99	-1.1	-1	-0.92	-0.99
1e-01	-1	-1.1	-1	-0.91	-1.1
1e+00	-1.3	-0.91	-0.92	-1	-1
1e+01	-0.75	-0.92	-0.91	-1	-1
1e+02	-0.76	-0.93	-0.91	-1	-1
1e+03	-0.75	-0.96	-0.92	-1	-1
1e+04	-0.78	-0.99	-0.95	-1	-1
1e+05	-1	-1	-1	-0.99	-1

method names gives the type of boundary condition used, e.g., BOXL1 corresponds to the BOXL methods using boundary conditions of type 1.

The bars on the accuracy plots are obtained by considering the different results for each $k_{F,i}$, $i = 1, \ldots, 11$. The central points on the errorbars correspond to the average over all $k_{F,i}$, and the extrema on the bars corresponds to the minimum and maximum value over all $k_{F,i}$. The results show again that the box method is the most accurate for pressure, whereas the MPFA method is most accurate for the flux. However, there are relatively large variation over the different fracture permeabilities.

The convergence rate for pressure was second order (-1 in DOFs) for the BOXQ1, BOXQ3, MPFAQ1, and MPFAQ3 methods. However, for the other methods the convergence rate depended on the fracture permeability as shown in Table 3.

The convergence rates for flux are shown in Table 4. All the BOXL methods showed a similar behavior; we only show the BOXL2 method in Table 4. The BOXQ methods showed a similar behavior as the BOXQ2 method in Table 4, and all the MPFAQ methods all showed different behavior, also see Table 4.

4.3. **Accuracy of the BOXL method.** Note that the convergence rates for the BOXL method in Tables 3 and 4 were obtained at the last, i.e. fifth, refinement level of the grids in Figure 10. For this grid the error in the BOXL method was larger than approximately 1e-5. However, for finer grids we observed a lack of convergence for the BOXL method. The convergence for the BOXL method ceased when the errors got smaller than approximately 1e-5.

To explain this, we consider the square domain of size $d \times d$ in Figure 12. For a homogeneous scalar permeability $\boldsymbol{K}(x, y) = 1$ the flux stencil for the central vertex in the figure (corresponding to mass conservation for the

TABLE 4. Convergence rates for flux in terms of DOFs. Convergence rates for the BOXL1, BOXL3, BOXQ1, and BOXQ3 methods are not shown here, but discussed in the main text.

Perm.	BOXL2	BOXQ2	MPFAQ1	MPFAQ2	MPFAQ3
1e-05	-0.75	-0.51	-0.94	-0.71	-0.85
1e-04	-0.75	-0.51	-0.94	-0.71	-0.85
1e-03	-0.75	-0.51	-0.93	-0.71	-0.83
1e-02	-0.75	-0.51	-0.94	-0.73	-0.86
1e-01	-0.75	-0.51	-0.94	-0.74	-0.88
1e+00	-0.76	-0.51	-0.93	-1	-1
1e+01	-0.76	-0.52	-0.93	-0.77	-0.84
1e+02	-0.91	-0.71	-0.87	-0.9	-0.83
1e+03	-1.4	-1.2	-0.95	-1	-0.96
1e+04	-1.4	-1.3	-0.99	-1	-1
1e+05	-1.3	-1.3	-0.99	-1	-1

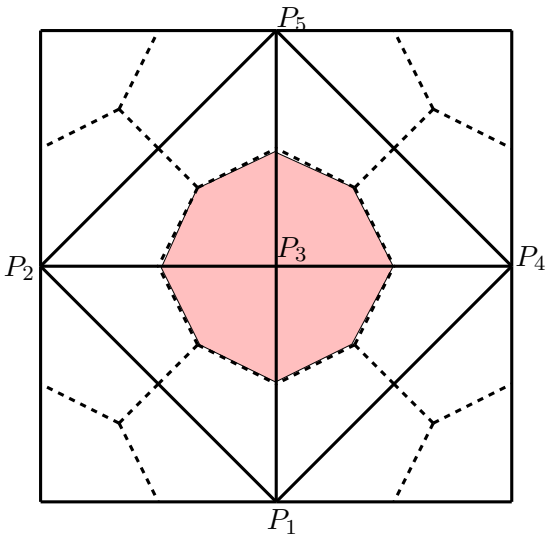


FIGURE 12. A square domain of size $d \times d$, discretized with 8 triangles.

shaded control volume) is given by

$$-P_1 - P_2 + 4P_3 - P_4 - P_5 = 0. \tag{25}$$

Consider now a horizontal (lower dimensional) fracture of aperture ϵ extending from P_2 to P_4 in the figure. Let the permeability of the fracture be $k_F = 1$. Then the flux stencil for the BOXL method becomes

$$-P_1 - (1 + \frac{2\epsilon}{d})P_2 + (4 + \frac{4\epsilon}{d})P_3 - (1 + \frac{2\epsilon}{d})P_4 - P_5 = 0.$$

The introduction of the additional terms in the last equation, introduces an error of size $\mathcal{O}(\epsilon/d)$ into the system matrix for the BOXL method. This error is not reduced as the grid is refined, and explains why the BOXL cannot converge.

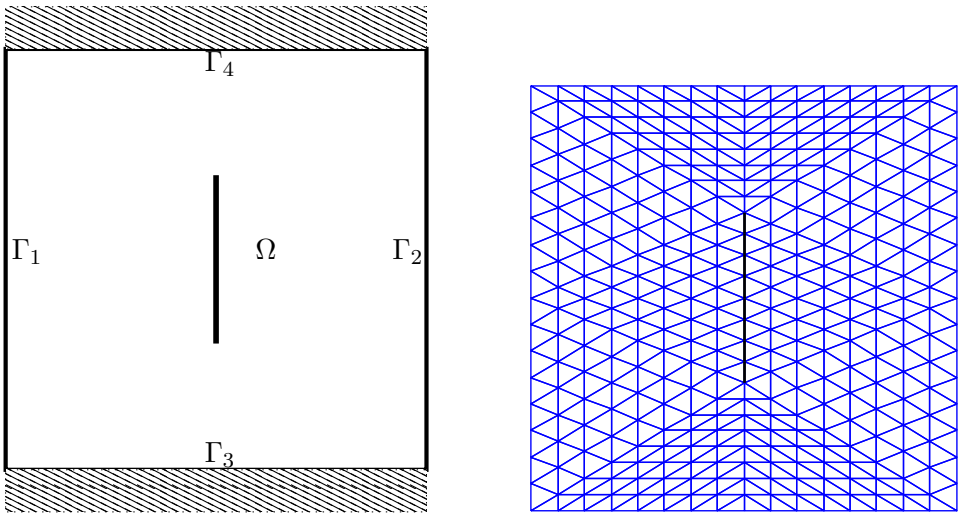


FIGURE 13. Left: A square domain with a vertical fracture. Right: Initial discretization (Ref0) of the domain to the left.

4.4. **Low-permeable fractures.** In this and the following sections, we will use boundary conditions of type-3, but they will not be determined from an analytical solution as in the previous sections, rather we solve the following problem:

$$\nabla \cdot (\boldsymbol{K} \nabla P) = 0, \qquad \text{in } \Omega \tag{26a}$$

$$P = 1, \qquad \text{on } \Gamma_1 \tag{26b}$$

$$P = 0, \qquad \text{on } \Gamma_2 \tag{26c}$$

$$(\boldsymbol{K} \nabla P) \cdot \boldsymbol{n} = 0, \qquad \text{on } \Gamma_3 \cup \Gamma_4. \tag{26d}$$

The domain $\Omega = [0,1] \times [0,1]$ is shown in Figure 13 (left). The boundary of the domain consists of the union of the four straight lines, Γ_i, $i = 1, \ldots, 4$, shown in Figure 13 (left).

We will assume that the permeability tensor is a piecewise constant scalar given as

$$\boldsymbol{K}(\boldsymbol{x}) = \begin{cases} k_M, & \boldsymbol{x} \text{ in matrix,} \\ k_F, & \boldsymbol{x} \text{ in fractures} \end{cases} \tag{27}$$

We will show that the BOXL method should not be used when fractures are barriers to flow, i.e. for the case $k_F < k_M$. Consider a vertical fracture (drawn in black) located in the center of the domain, see Figure 13 (right). We will assume a matrix permeability of k_M=1, and the fracture permeability $k_F < 1$. The the domain is discretized as shown in Figure 13 (right), and a fracture width of ϵ=1e-4 is assumed. We first solve Problem (26) with the MPFAQ method, using a fracture permeability of k_F=1e-10. The result is shown in Figure 14 (left). Here we have traced 40 streamlines for the pressure solution. Refer to [38] for details about the streamline tracing algorithm. We see that the flow correctly avoids the barrier. We now solve the same problem using the BOXL method. The result is shown in Figure 14 (right). We see that the barrier is not respected by the box solution. The

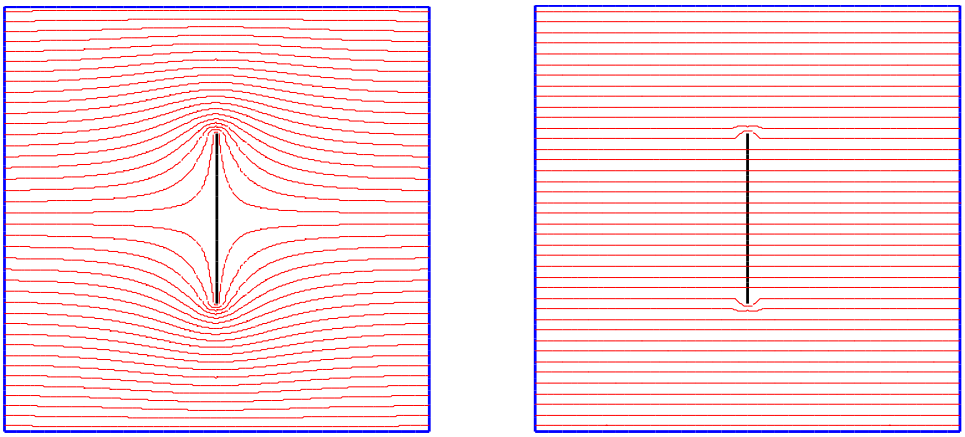


FIGURE 14. Streamlines in a domain with a low permeable vertical fracture. Left: MPFAQ method, Right: BOXL method

streamlines penetrates the barrier as it did not exist. Note that in order to trace streamlines with the box solution, a flux recovery procedure was used, see [38] for details.

The behavior of the box method can be explained by considering Figure 12. Assume first a homogeneous domain, i.e. $k_F = k_M = 1$. Then the flux stencil for the central vertex in the figure is given by Equation (25) Consider now a vertical fracture of aperture ϵ extending from P_1 to P_5 in the figure. Let the permeability of the fracture be k_F, and the permeability of the matrix be k_M=1. Then the flux stencil for the box method becomes

$$-P_2 - (1 + \frac{2k_F\epsilon}{d})P_1 + (4 + \frac{4k_F\epsilon}{d})P_3 - (1 + \frac{2k_F\epsilon}{d})P_5 - P_4 = 0.$$

As k_F approaches zero, the last expression approaches Equation (25). Hence, the influence of a low permeable fracture ($k_F < k_M$) on the flow behavior will be neglected by the box method, see Figure 14 (right).

We note that the type of boundary conditions can make the error in the BOXL method less noticeable. This can be seen in the results from Section 4.2 in Tables 3 and 4. Here, it seems like the BOXL method converges for the low permeable cases. However, the behavior of the system is here dominated by the boundary conditions, and the true solution does not change significantly from the homogeneous case to the case where k_F=1e-5.

4.5. **Investigation of the condition number.** The condition number of a matrix $\boldsymbol{A}$ is defined as

$$\kappa = \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\|,$$

where the norm is usually taken to be the 2-norm. The condition number is important due to its relation to the propagation of round-off errors and to the convergence rate of an iterative solver [36, 39, 63].

In our case, the condition number of the system matrix resulting from discretizing the problem in Equation (26a), depends on many factors. Firstly, the condition number depends on the number of degrees of freedom (DOFs)

TABLE 5. Number of unknowns and condition number for each method as a function of grid refinement level.

		Ref0	Ref1	Ref2	Ref3	Ref4	Ref5
DOF	BOX	3	15	63	255	1023	4095
	MPFA	8	32	128	512	2048	8192
COND	BOX	3.7	15	58	220	855	3371
	MPFA	9	31	120	479	1919	7683

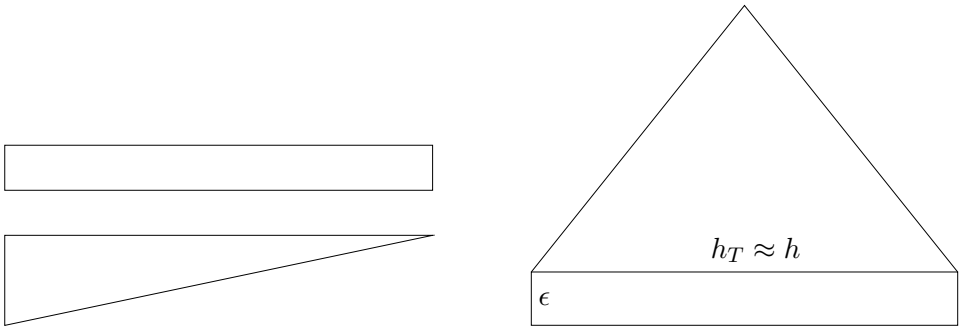


FIGURE 15. Left: A "thin" rectangle and a triangle with a small angle. Such "thin" elements tend to cause ill-conditioned systems. Right: Illustration of the parameters ϵ and h. Two elements from a larger grid are shown. The triangle is a matrix element, and the rectangle is a fracture element. The side length h_T of the triangle is approximately the same size as h. Note that h_T is equal to the "length" of the fracture element.

in the solution. To illustrate this, consider the case of a homogeneous domain with no fractures and permeability $\boldsymbol{K} \equiv 1$. A sequence of grids with increasing DOFs is obtained starting with a grid similar to that shown in Figure 7 (left). The refinements are obtained in the same way as explained in Section 4.1, without random perturbation of the corners. The results for the condition number in Table 5 reveal an almost exact linear relation on the form

$$\kappa = aN + b,$$

where N is the number of degrees of freedom, and a and b are constants. A least-square fit to the data shows that $a \approx .822$ for the box method, $a \approx 1.10$ for the MPFA method.

In equi-dimensional fracture models, "thin" elements (see Figure 15 (left)) typically occur in the fracture. It is well known that for triangular grids, small angles tend to produce a high condition number. For quadrilaterals, "thin" elements, e.g. a rectangle with one side length much larger than the other side length, may also cause problems. A common way to eliminate "thin" elements is to use adaptive grid refinement. Drawbacks of the adaptive method may be an increased number of unknowns, and more difficult mesh generation. In this paper we will not consider the adaptive method further.

The fracture aperture ϵ and the grid size parameter h are illustrated in Figure 15 (right). Note that for a coarse mesh, we typically have $h \gg \epsilon$, and the elements in the fractures tend to be very "thin". However, as the grids are refined and ϵ is kept fixed, h will approach ϵ in size, and the fracture elements will become more and more regular in shape. This effect tends to reduce the condition number, but since the condition number also increase with the number of degrees of freedom, or with the grid refinement level, there is a trade-off here.

The parameters h and ϵ, are not independent in their influence on the condition number. If $h \gg \epsilon$ the condition number only depends on the single parameter $\alpha = h/\epsilon$ (all other parameters held fixed). This is also true for the fracture permeability k_F and the matrix permeability k_M, which can be substituted for by the single parameter $\beta = k_F/k_M$. At last, it was found that for the box method all of the previous parameters could be substituted for by the single parameter

$$\gamma = \frac{hk_F}{\epsilon k_M}$$

Hence the influence of the different parameters on the condition number for a given grid $\mathcal{G}$ can be summarized as

$$\kappa = \begin{cases} f_M^{\mathcal{G}}(\gamma, N), & M\text{=BOXL, BOXQ, BOXT} \\ f_M^{\mathcal{G}}(\alpha, \beta, N), & M\text{=MPFAQ, MPFAT} \end{cases} \tag{28}$$

for an unknown function $f_M^{\mathcal{G}}$ and number of degrees of freedom N. Here $\mathcal{G}$ refers to a grid with a particular fracture configuration.

To sum up, if we also allow the grid parameter $\mathcal{G}$ to vary, the following factors may have an influence on the condition number:

- the parameters α, β, and γ
- number of fractures
- number of crossing fractures,
- fractures ending at the boundary of the domain, and
- how fracture endpoints are discretized.

In theory, the effects of each parameter on the condition number could be determined by looking at the coefficient matrix $\boldsymbol{A}$ and how parameters enter the matrix, and relations between the different elements. However, using this approach we were only able to discover the general relation in Equation (28); the form of the function $f_M^{\mathcal{G}}$ in that equation could only be determined from extensive numerical simulations for given fracture configuration $\mathcal{G}$ and selected values of the parameters α, β and γ.

The influence of β and γ is in the following investigated by changing the fracture permeability k_F. We consider 14 different test cases, as shown in Figure 16, corresponding to different number of fractures crossing each other at a point. For each grid we solve Equation (26a), with $k_M \equiv 1$. We also tried to improve the condition number for the MPFA method by using a simple diagonal scaling preconditioner. The scaling was done by dividing each row of the system matrix by the maximum element in absolute value in that row (typically the diagonal element). The effect of the scaling is to normalize the coefficient matrix, which may be efficient in the case where

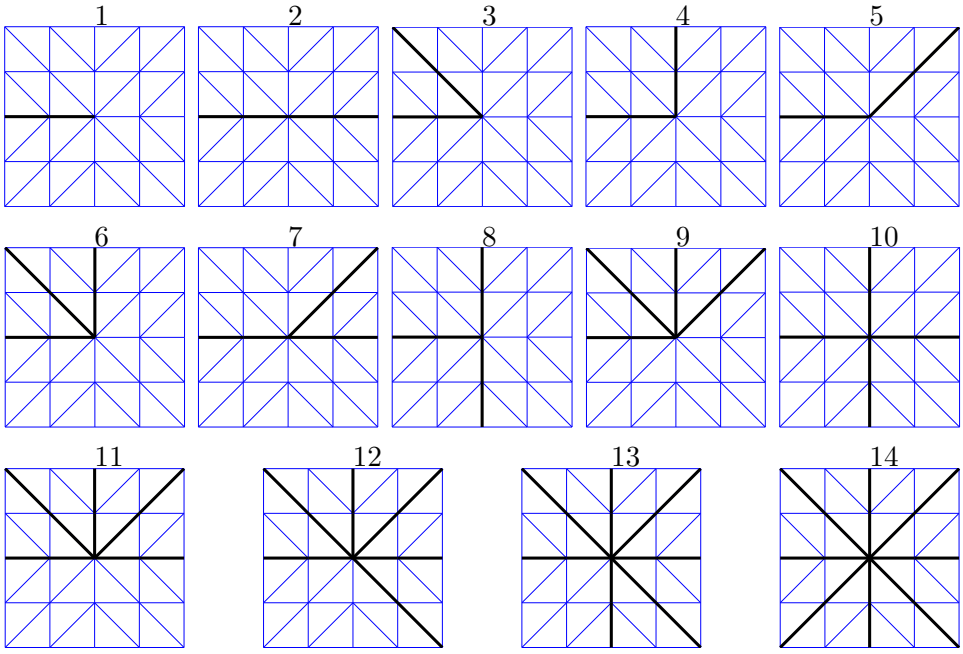


FIGURE 16. 14 different test cases.

the elements of each row considered separately are of approximately the same magnitude, but at the same time, relatively large differences between different rows exists. In order to keep the symmetry, the system matrix of the box method was not scaled.

The results for the condition number as a function of fracture permeability is shown in Figure 17. In the test, the grids in Figure 16 were refined one level, such that the DOFs of the box methods were approximately 65, and the number of degrees of freedom for the MPFA methods were around 130. It was found that the condition number for the BOXQ and BOXT methods could not be distinguished visually on the figures. Hence, the condition number of these methods are represented by the BOXE (box equi-dimensional) curve in the figure. Also, several of the cases showed very similar behavior for the condition number, e.g., case-1 and case-2 were similar, and are represented by the single subfigure with title "case1-2". In addition, the condition number for the BOXL method is only calculated for the cases $k_F > k_M = 1$, due to its deficiency for the low permeable cases, confer with the discussion in Section 4.4.

The test showed that the condition number for the BOXE and BOXL methods for all cases increased steadily with fracture permeability when fracture permeability was larger than approximately 10^{-3}, see Figure 17. For low fracture permeability, the condition number was low for all equi-dimensional methods for cases 1-5, cases 7-8, and case 10. The MPFA methods had low condition number for all the high-permeable cases. We also note that the BOXL method was better conditioned than the BOXE method for all the high-permeable cases.

4.6. **Convergence study for a synthetic test case.** In this section we show how the methods perform on a more realistic test case. We compare

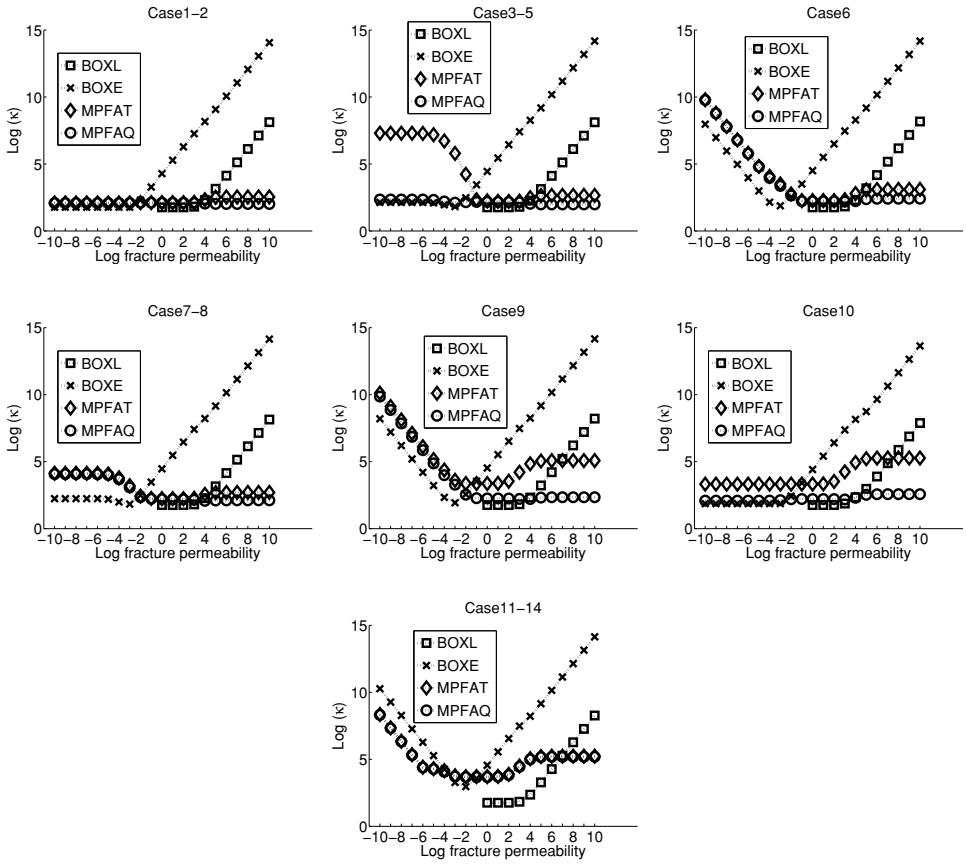


FIGURE 17. The condition number for the 14 different test cases.

the BOXL method and the MPFAQ method on a coarse and a fine grid of a fracture realization, see Figure 18 (top). The coarse grid consist of 163 elements, and the fine grid has 8211 elements. We solve the boundary problem given in Equation (26), and use a fracture permeability of $k_F = 10^3$, a matrix permeability of $k_M \equiv 1$, and a fracture aperture of $\epsilon = 10^{-4}$.

A number of 40 streamlines was traced for each method on each grid. The streamlines for the BOXL method are red, and streamlines for the MPFAQ method are black. The streamlines are distributed uniformly on the left hand edge according to the inflow flux, and hence started at exactly the same points for each method.

We see that the streamlines converge as the grid is refined.

5. Summary and conclusions

In this paper we have compared discretization methods for a discrete fracture model in 2D. Two conceptually different finite volume methods were considered for the pressure equation: A vertex-centered box method based on a Donald-mesh, and a cell-centered MPFA O-method. We considered accuracy and numerical convergence of flux and pressure for homogeneous

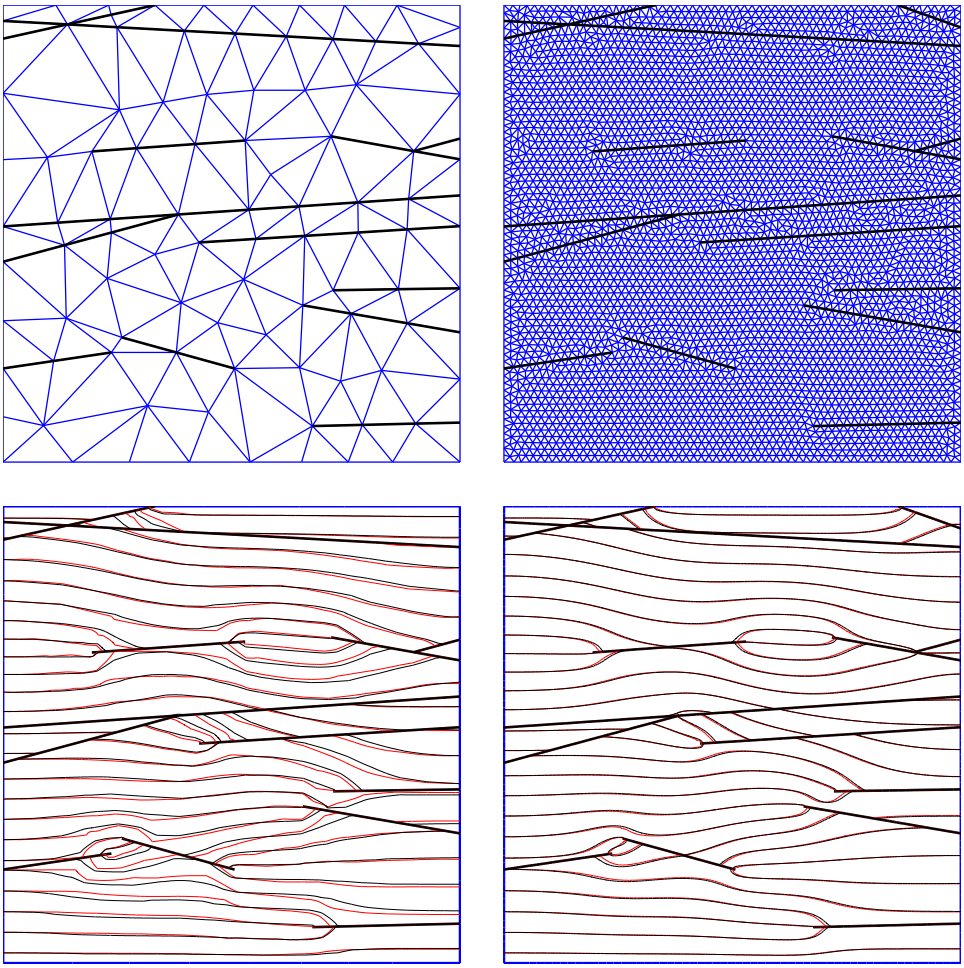


FIGURE 18. Grids and streamlines for the synthetic test case. Top Left: Coarse grid. Top Right: Fine grid. Bottom Left: Streamlines for the coarse grid. Bottom Right: Streamlines for the fine grid.

and fractured systems. In addition, the effects of fracture aperture, permeability contrast, and number of crossing fractures, on the condition number were investigated.

The MPFA method used an equi-dimensional treatment of the fractures, whereas the box method was tested with both lower-dimensional and equi-dimensional fractures. A procedure for expanding a lower dimensional discrete fracture-model to an equi-dimensional model was proposed.

The box methods produced a symmetric positive definite matrix, which enabled a fast and efficient solution of the corresponding linear system. The MPFA method produced an asymmetric matrix which excluded the use of high-efficient solvers. The box method were in general the most accurate methods for the pressure, whereas the MPFA-methods were the most accurate methods for the flux variable.

The lower-dimensional box method could only be used for high-permeable fractures. Also, this method could not achieve high-accuracy results, due

to the simplified treatment of the fractures. However, for most cases, the high-accuracy deficiency will probably not be an issue due to coarse meshes or other more important errors.

Several relations between different parameters in the fracture model, and the condition number was found. The numerical tests indicated that the MPFA-methods have well-behaved condition number for the case of high-permeable fractures in low-permeable matrix.

Acknowledgment. This work was supported by the International Research Training Group "Non-Linearities and Upscaling in Porous Media (NUPUS)" financed by the German Research Foundation (DFG).

References

[1] I. Aavatsmark. An introduction to multipoint flux approxmations for quadrilateral grids. *Comput. Geosci.*, 6:405–432, 2002.

[2] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth. Discretization on unstructured grids for inhomogeneous, anisotropic media. part i: Derivation of the methods. *SIAM Journal of Scientific Computing*, 19(5):1700–1716, 1998.

[3] I. Aavatsmark and G.T. Eigestad. Numerical convergence of the MPFA O-method and U-method for general quadrilateral grids. *Int. J. Numer. Meth. Fluids*, 51:939–961, 2006.

[4] P.M. Adler and J.-F. Thovert. *Fractures and fracture networks*, volume 15 of *Theory and applications of transport in porous media*. Kluwer Academic Publ., Boston, 1999.

[5] T. Arbogast, C.N. Dawson, P.T. Keenan, M.F. Wheeler, and I. Yotov. Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comput.*, 19(2):404–425, 1998.

[6] T. Arbogast, M.F. Wheeler, and I. Yotov. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite-differences. *SIAM J. Numer. Anal.*, 34(2):828–852, 1997.

[7] D.N. Arnold, D. Boffi, and R.S. Falk. Quadrilateral H(div) finite elements. *SIAM J. Numer. Anal.*, 42(6):2429–2451, 2005.

[8] A. Assteerawatt, R. Helmig, H. Hægland, A. Bárdossy, and H.K. Dahle. Simulation of flow and transport processes in a discrete fracture-matrix system. i. geostatistical generation of fractures on an aquifer analogue scale. Submitted to *Water Resources Research*, 2008.

[9] B.R. Baliga and S.V. Patankar. A new finite-element formulation for convection-diffusion problems. *Numerical Heat Transfer*, 3:393–409, 1980.

[10] R.E. Bank and D.J. Rose. Some error estimates for the box method. *SIAM J. Numer. Anal.*, 24(4):777–787, 1987.

[11] J.A. Barker. Transport in fractured rock. In R.A. Downing and W.B. Wilkinson, editors, *Applied groundwater hydrology*, Oxford science publications, pages 199–216. Clarendon Press, Oxford, 1991.

[12] J. Bear. Modeling flow and contaminant transport in fractured rocks. In J. Bear, C.-F. Tsang, and G. de Marsily, editors, *Flow and contaminant transport in fractured rock*, pages 1–37. Academic Press, San Diego, 1993.

[13] M. Belayneh, S. Geiger, and S.K. Matthäi. Numerical simulation of water injection into layered fractured carbonate reservoir analogs. *AAPG Bulletin*, 90(10):1473–1493, 2006.

[14] B. Berkowitz. Characterizing flow and transport in fractured geological media: A review. *Adv. Water Resour.*, 25:861–884, 2002.

[15] E. Bonnet, O. Bour, N.E. Odling, P. Davy, I. Main, P. Cowie, and B. Berkowitz. Scaling of fracture systems in geological media. *Reviews of Geophysics*, 39(3):347–383, 2001.

[16] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer series in computational mathematics, 15. Springer-Verlag, New York, 1991.

[17] Z. Cai, J.E. Jones, S.F. McCormick, and T.F. Russell. Control-volume mixed finite element methods. *Comput. Geosci.*, 1:289–315, 1997.
[18] Z. Cai, J. Mandel, and S. McCormick. The finite volume element method for diffusion equations on general triangulations. *SIAM J. Numer. Anal.*, 28(2):392–402, 1991.
[19] Y. Caillabet, P. Fabrie, P. Landereau, B. Noetinger, and M. Quintard. Implementation of a finite-volume method for the determination of effective parameters in fissured porous media. *Numer. Meth. Part. D.E.*, 16:237–263, 2000.
[20] P. Castillo. A review of the local discontinuous Galerkin (LDG) method applied to elliptic problems. *Appl. Numer. Math.*, 56:1307–1313, 2006.
[21] Z. Chen. On the control volume finite element methods and their application to multiphase flow. *Networks and Heterogeneous Media*, 1(4):689–706, 2006.
[22] P.G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 1980.
[23] Committee on Fracture Characterization and Fluid Flow, U.S. National Commitee for Rock Mechanics, Geotechnical Board, Board on Energy and Environmental Systems, Commission on Engineering and Technical Systems, and National Research Council. *Rock fractures and fluid flow : contemporary understanding and applications*. National Academy Press, Washington, D.C., 1996.
[24] A. Datta-Gupta and M.J. King. *Streamline simulation: Theory and practice*. SPE Text book, 2007.
[25] T. Davis. Comparison of CHOLMOD and the preconditioned conjugate gradient method. Available online: http://www.cise.ufl.edu/research/sparse/pcg_compare/, 2007.
[26] T.A. Davis. Umpfpack, unsymmetric multifrontal method. http://www.cise.ufl.edu/research/sparse/umfpack/, 2008.
[27] P. Dietrich, R. Helmig, M. Sauter, H. Hötzl, J. Köngeter, and G. Teutsch, editors. *Flow and transport in fractured porous media*. Springer, Berlin, 2005.
[28] M.G. Edwards and M. Pal. Positive-definite q-families of continuous sub-cell Darcy-flux CVD(MPFA) finite volume schemes and the mixed finte element method. *Int. J. Numer. Meth. Fluids*, 57:355–387, 2008.
[29] M.G. Edwards and C.F. Rogers. Finite volume discretization with imposed flux continuity for the general tensor pressure equation. *Computational Geosciences*, 2:259–290, 1998.
[30] G.T. Eigestad, I. Aavatsmark, and M. Espedal. Symmetry and M-matrix issues for the O-method on an unstructured grid. *Comput. Geosci.*, 6:381–404, 2002.
[31] G.T. Eigestad and R.A. Klausen. On the convergence of the multi-point flux approximation O-method: Numerical experiments for discontinuous permeability. *Numer. Meth. Part. D.E.*, 21(6):1079–1098, 2005.
[32] R.E. Ewing, T. Lin, and Y. Lin. On the accuracy of the finite volume element method based on piecewise linear polynomials. *SIAM J. Numer. Anal.*, 39(6):1865–1888, 2002.
[33] R. Eymard, T. Gallouët, and R. Herbin. Finite volume methods. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of numerical analysis, vol. VII*. North-Holland, Amsterdam, 2000.
[34] A. Fuchs. *Optimierte Delaunay-Triangulierungen zur Vernetzung getrimmter NURBS-Körper*. PhD thesis, Stuttgart University, 1999.
[35] S. Gebauer, L. Neunhäuserer, R. Kornhuber, S. Ochs, R. Hinkelmann, and R. Helmig. Equidimensional modelling of flow and transport processes i fractured porous systems i. In Proc [56].
[36] G.H. Golub and C.F. Van Loan. *Matrix computations*. Johns Hopkins studies in the mathematical sciences ; 3. Johns Hopkins University Press, Baltimore, third edition, 1996.
[37] H. Hægland. *Streamline methods with application to flow in fractured media*. PhD thesis, Dept. of Mathematics, University of Bergen, Norway, 2009. In preparation.
[38] H. Hægland, A. Assteerawatt, R. Helmig, and H.K. Dahle. Simulation of flow and transport for a discrete fracture matrix system ii. effcient and accurate streamline approach. Submitted to *Water Resources Research*, 2008.

[39] N.J. Higham. *Accuracy and stability of numerical algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, second edition, 2002.
[40] R. Holm, R. Kaufmann, B.-O. Heimsund, E. Øian, and M.S. Espedal. Meshing of domains with complex internal geometries. *Numerical Linear Algebra with Applications*, 13(9):717–731, 2006.
[41] H. Hoteit and A. Firoozabadi. Multicomponent fluid flow by discontinuous Galerkin and mixed methods in unfractured and fractured media. *Water Resour. Res.*, 41(W11412, doi:10.1029/2005WR004339), 2005.
[42] P.S. Huyakorn and G.F. Pinder. *Computational methods in subsurface flow*. Academic Press, New York, 1983.
[43] J. Hyman, M. Shashkov, and S. Steinberg. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. *J. Comput. Phys.*, 132:130–148, 1997.
[44] R. Juanes, J. Samper, and J. Molinero. A general and efficient formulation of fractures and boundary conditions in the finite element method. *Int. J. Numer. Methods Eng.*, 54:1751–1774, 2002.
[45] M. Karimi-Fard, L.J. Durlofsky, and K. Aziz. An efficient discrete fracture model applicable for general purpose reservoir simulators. *SPE J.*, pages 227–236, June 2004.
[46] J.-G. Kim and M.D. Deo. Finite element, discrete-fracture model for multiphase flow in porous media. *AIChE Journal*, 46(6):1120–1130, 2000.
[47] R.A. Klausen and R. Winther. Robust convergence of multi point flux approximation on rough grids. *Numer. Math.*, 104:317–337, 2006.
[48] R. Li, Z. Chen, and W. Wu. *Generalized difference methods for differential equations*. Marcel Dekker, New York, 2000.
[49] V. Martin, J. Jaffré, and J.E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691, 2005.
[50] K.W. Morton and E. Süli. Finite volume methods and their analysis. *IMA Journal of Numerical Analysis*, 11:241–260, 1991.
[51] S.P. Neuman. Trends, prospects and challenges in quantifying flow and transport through fractured rocks. *Hydrogeology Journal*, 13:124–147, 2005.
[52] L. Neunhäuserer. *Diskretisierungsansätze zur Modellierung von Strömungs- und Transportprozessen in geklüftet-porösen Medien*. PhD thesis, University of Stuttgart, 2003.
[53] L. Neunhäuserer, S. Gebauer, S. Ochs, R. Hinkelmann, R. Kornhuber, and R. Helmig. Equidimensional modelling of flow and transport processes in fractured porous systems ii. In Proc [56].
[54] S.O. Ochs, R. Hinkelmann, N. Neunhäuserer, M. Suess, R. Helmig, S. Gebauer, and R. Kornhuber. Adaptive methods for the equidimensional modelling of flow and transport processes in fractured aquifers. In *5th International Conference on Hydro-Science & -Engineering, Warsaw, Poland*, 2002.
[55] C.F. Pinder, P.S. Huyakorn, and E.A. Sudicky. Simulation of flow and transport in fractured porous media. In J. Bear, C.-F. Tsang, and G. de Marsily, editors, *Flow and contaminant transport in fractured rock*, pages 395–435. Academic Press, San Diego, 1993.
[56] *Proceedings of the XIV International Conference on Computational Methods in Water Resources, vol. 1, 23-28 June, Delft, The Netherlands*, 2002.
[57] K. Pruess, J.S.Y. Wang, and Y.W. Tsang. On thermohydrologic conditions near high-level nuclear wastes emplaced in partially saturated fractured tuff. 1. simulation studies with explicit concideration of fracture effects. *Water Resour. Res.*, 26(6):1235–1248, 1990.
[58] V. Reichenberger, H. Jakobs, P. Bastian, and R. Helmig. A mixed-dimensional finite volume method for two-phase flow in fractured porous media. *Adv. Water Resour.*, 29(7):1020–1036, 2006.
[59] V. Reichenberger, H. Jakobs, P. Bastian, R. Helmig, and J. Niessner. Complex gas-water process in discrete fracture-matrix systems. Project report, 2006.

[60] M. Sahimi. *Flow and transport in porous media and fractured rock: From classical methods to modern approaches*. VCH, Weinheim, 1995.
[61] M. Shashkov and S. Steinberg. *Conservative finite-difference methods on general grids*. Symbolic and numeric computation series. CRC Press, Baca Raton, 1996.
[62] A. Silberhorn-Hemminger. *Modellierung von Kluftaquifersystemen: Geostatistische Analyse und deterministisch-stochastische Kluftgenerierung*. PhD thesis, University of Stuttgart, 2002.
[63] G.W. Stewart and J.-G. Sun. *Matrix pertubation theory*. Computer science and scientific computing. Academic press, Boston, 1990.
[64] G. Strang and G.J. Fix. *An analysis of the finite element method*. Prentice-Hall series in automatic computation. Prentice-Hall, Englewood Cliffs, N.J., 1973.
[65] *SPE Reservoir Simulation Symposium, Dallas, Texas, 8-11 June*, 1997.
[66] A. Teimoori, Z. Chen, S.S. Rahman, and T. Tran. Effective permeability calculation using boundary element method in naturally fractured reservoirs. *Petroleum Science And Technology*, 23:693–709, 2005.
[67] S. Verma and K. Aziz. A control-volume scheme for flexible grids in reservoir simulation, SPE 37999. In Symp [65].
[68] E.W. Weisstein. "Plane-Plane Intersection." from MathWorld–A wolfram web resource. http://mathworld.wolfram.com/Plane-PlaneIntersection.html.
[69] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The finite element method : its basis and fundamentals*. Elsevier, Amsterdam, 6th edition, 2005.

Paper F

Simulation of flow and transport in discrete fracture-matrix system: II. Efficient and accurate streamline approach

*

* Submitted to *Water Resources Research*, 2008

SIMULATION OF FLOW AND TRANSPORT PROCESSES IN A DISCRETE FRACTURE-MATRIX SYSTEM II. EFFICIENT AND ACCURATE STREAMLINE APPROACH

H. HÆGLAND, A. ASSTEERAWATT, R. HELMIG, AND H. K. DAHLE

Abstract. Simulations of flow and transport in fractured porous rocks using a discrete fracture model have gradually become more practical, as a consequence of increased computer power and improved simulation and characterization techniques. Fractures in a discrete model are generally described with one dimension less than the surrounding matrix, the so-called lower-dimensional approach. However, high numerical diffusion in the transport simulation causes an increased computational demand due to the fine-grid requirement. A streamline method for transport for a lower dimensional discrete fracture model (DFML) is proposed in this paper. By solving the mass conservation equation using a vertex-centered finite volume scheme, a pressure field is obtained. Then, a fracture expansion and a flux recovery method are carried out to determine new mass conservative fluxes on a hybrid grid of triangles and quadrilaterals, on which streamlines are traced. Only the advective transport is assumed for the streamline method. The results of the streamline method are compared with a grid-based finite volume method using two different fracture-matrix systems: simple systems (a single fracture or systematically distributed fractures) and complex fracture-matrix systems. Significantly different transport behavior can be observed in the two types of systems. The numerical diffusion in the grid-based transport simulation smears out the heterogeneity effect (fast transport in the fractures and slow transport in the matrix) and delays the plume migration. Whereas, the purely advective transport without numerical diffusion in the streamline method leads to faster transport.

1. Introduction

Simulations of flow and transport in fractured porous rocks using a discrete fracture model have gradually become more practical, as a consequence of increased computer power and improved simulation and characterization techniques. In a discrete fracture model, fractures may either be discretized with the same dimension as the surrounding matrix elements, the so-called equi-dimensional approach, or with one dimension less than the matrix, the so-called lower-dimensional approach. The comparison study of the two discretization approaches presented in Neunhäuserer [19] revealed a number of local differences for the flow and transport, but and only minor differences globally. Sufficient accuracy of global solutions with reduction of the computational time have lead to wide-spread application of the lower-dimensional approach, see e.g., [23, 17, 14, 16], and references therein.

In advective dominated problems like transport in fracture-matrix systems, grid-based methods such as finite difference, finite element and finite volume methods all using the Eulerian approach, suffer from numerical diffusion. High numerical diffusion in the transport simulation, as shown in the associated paper, gives rise to an increased computational burden since a very fine grid is required. Streamline methods have become a viable alternative to traditional finite element or finite difference reservoir simulation during the last decade ([15, 27]). The advantages of streamline simulation are lower computational demand and less numerical diffusion compared with a grid-based transport model. On

the reservoir scale where fractures and matrix are treated as two interacting continua, promising results from streamline tracing have been presented by Huang et al. [12] and Al-Huthali and Datta-Gupta [1]. Their results showed a close agreement with the results from a grid-based finite difference simulation with a significant reduction in run time. In this work, we propose a streamline method for transport for the lower dimensional discrete fracture model (DFML). Similar to the grid-based methods, the streamline method is based on the velocity field determined from a flow simulation.

The precision of streamline tracing strongly depends on the accuracy of the velocity field ([18]). For finite element based solutions, approximating the velocity from pressure gradients results in discontinuous fluxes at element boundaries and hence lack of mass conservation ([8]). Many papers have considered this problem recently, see e.g. [7, 5, 26], and references therein. Cordes and Kinzelbach [6] proposed an inexpensive technique for deriving a continuous distribution of fluxes from the finite element solutions. The method solves a local problem for each grid node to obtain conservative fluxes in a patch surrounding the node. This technique was extended by Prévost et al. [22] for the control-volume finite element scheme on unstructured grids. A flux continuous velocity for a sub cell of a control volume (triangular or quadrilateral in 2D, and tetrahedron or hexahedron in 3D) were reconstructed. In this work, a flux recovery for a two-dimensional fracture-matrix system based on the work of Cordes and Kinzelbach [6] and Prévost et al. [22] is introduced. Continuous and mass conservative fluxes for all sub cells of a control volume denoted as sub control-volumes are recorded and are later used for streamline tracing. Additionally, when streamline tracing is considered, lower-dimensional fractures, which are assumed in the flow simulation have to be extended to equi-dimensional fractures to obtain well-defined velocities in the fractures.

Due to the post processing and the use of unstructured grids, streamline tracing for general quadrilateral grids are required. For a regular quadrilateral mesh (rectangular mesh), Pollock's method [20] has been widely used. The extension of Pollock's method to unstructured grids has been presented in several studies. Cordes and Kinzelbach [6] extended Pollock's method to linear and bilinear finite element methods for groundwater flow, and later Prévost et al. [22] extended it for streamline tracing with the control volume finite element method, flux continuous scheme and the multipoint flux approximation (MPFA) method.

The objective of this paper is to present a streamline method for a lower dimensional discrete fracture model (DFML). In the next section, the streamline method is presented stepwise. First, the governing equation and numerical discretization for the flow process are summarized. Second, the flux recovery together with fracture expansion are described. Later, the streamline tracing using Pollock's method on unstructured grids and the evaluation of the breakthrough curve from the time-of-flight are introduced. Finally, the results obtained from the streamline method and from the grid-based finite volume method are compared.

2. Streamline Method

2.1. **Solution of the flow equation.** The continuity equation for an incompressible fluid in a nondeformable matrix is given as

$$\nabla \cdot \boldsymbol{q} = 0 \,, \tag{1}$$

where $\boldsymbol{q}$ is the Darcy velocity. Combining Equation (1) with Darcy's law and neglecting the gravitational effect yields

$$\nabla \cdot \boldsymbol{q} = -\nabla \cdot \frac{\boldsymbol{K}}{\mu} \nabla p = 0 \,, \tag{2}$$

where $\boldsymbol{K}$ is the permeability, μ is the dynamic viscosity and p is the pressure, [4].
A vertex-centered finite volume method, also called box method, is used in this study since it can be applied to unstructured grids of a fracture-matrix system and is locally mass conservative ([23]). The spatial discretization of the box method, described in more detail in Hægland et al. [10], is based on a primary finite element (FE) mesh and a secondary finite volume (FV) mesh. First, the domain is discretized into a FE mesh, where matrix properties are assumed to be constant on each element. Next, by connecting element barycenters with edge midpoints, a FV mesh is constructed, see Figure 1. With each vertex of the FV mesh there is an associated control volume. Each control volume consists of a set of sub cells denoted sub control-volumes as illustrated in Figure 1. Fractures are described by a one-dimensional line in a two-dimensional domain with an associated virtual width equal to their aperture, the so-called lower-dimensional approach ([23]).

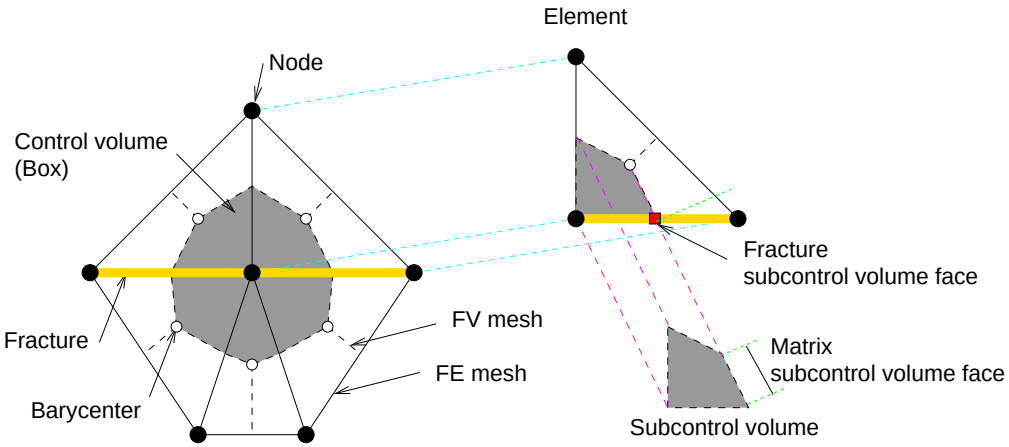
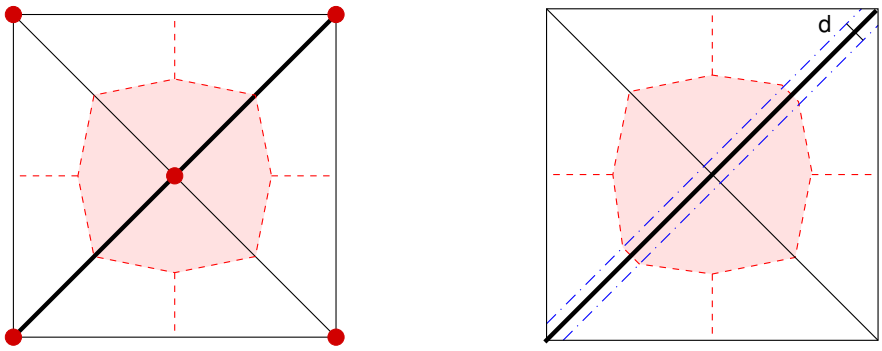


FIGURE 1. Finite Element (FE) and Finite Volume (FV) mesh.

The pressure is assumed to vary linearly within each element, and fluxes are evaluated using Darcy's law at the exterior boundary of a control volume. denoted as sub control-volume faces (see Figure 1). After the linear system of equations for the pressure field has been solved, conservative fluxes over all control volume faces can be determined. Note that fluxes are in general discontinuous at element boundaries, but they are continuous at the control volume boundaries.

2.2. **Flux recovery.** The precision of streamline tracing strongly depends on the accuracy of the velocity field. Approximating the velocity field using pressure gradients from the flow simulation results in discontinuous fluxes at element boundaries, not only when the permeabilities of neighboring elements are different but also when they are the same ([25]). On the basis of Cordes and Kinzelbach [6] and Prévost et al. [22], a flux recovery for a two-dimensional fracture-matrix system is introduced in this work to obtain continuous fluxes on a sub-quadrilateral grid. Additionally, lower-dimensional fractures, which are assumed in the flow simulation, have to be extended to equi-dimensional fractures to obtain well-defined velocities in the fractures (not only parallel to the fractures orientation), see [10].

Fractures are expanded such that the resulting 2D fractures have a width equal to the associated fracture aperture d and the 1D fracture is the center line, see Figure 2.



(a) Lower-dimensional 1D fracture

(b) Equi-dimensional 2D fracture

FIGURE 2. Expanding a 1D fracture with associated aperture d to a 2D fracture.

Figure 3 shows the five cases that are most likely to occur in a discretized fracture-matrix system. The control volumes are classified as: (type 1) no fractures, (type 2) a single crossing fracture, (type 3) two crossing fractures, (type 4) a single ending fracture and (type 5) an ending fracture and one passing through. The flux recovery for control volumes is presented here in detail for type 2, and is briefly discussed for the other types afterwards. The expansion procedure for the general case is discussed in Hægland [10].

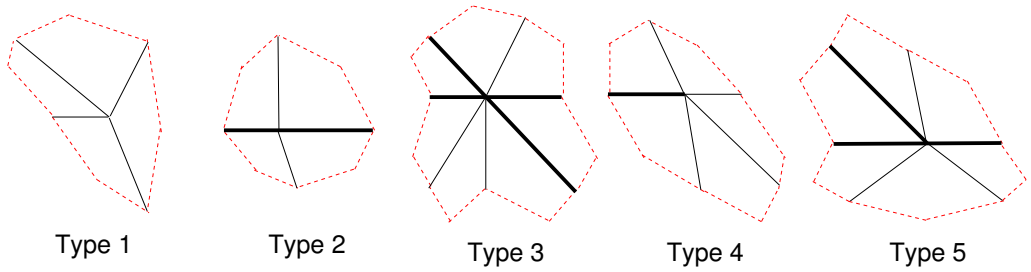
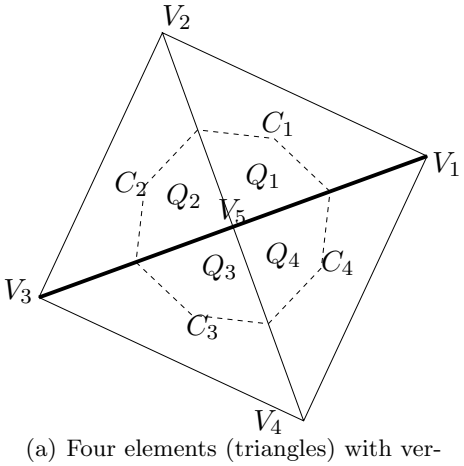
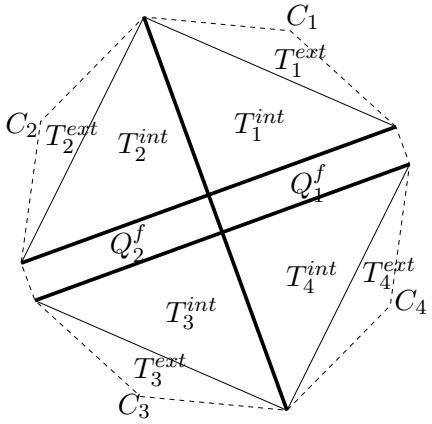


FIGURE 3. The different types of control volumes in a discretized fracture-matrix system. Fractures are indicated with heavy lines.

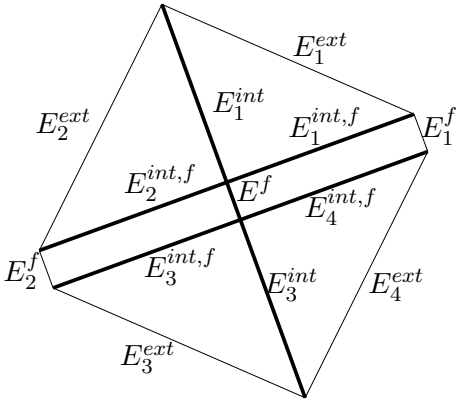
A simple prototype control volume of type 2 is shown in Figure 4(a). The control volume Ω_{cv} of the central vertex V_5, indicated with the dashed line, is the union of parts of four triangular elements together with a segment of a single fracture, indicated by the heavy line. The control volume comprises a total of N quadrilateral sub control-volumes denoted Q_j, where $j = 1, \ldots, N$ (in this simple case $N = 4$). We order Q_j counter clockwise such that the common edge between Q_1 and Q_N coincides with the fracture. The 1D fracture is expanded to a 2D fracture with aperture d, such that two new rectangular sub control-volumes Q_1^f and Q_2^f are created inside the control volume, see Figure 4(b). Each quadrilateral Q_j is split into two triangles, one interior T_j^{int} and one exterior T_j^{ext}, relative to the central node of the control volume, see Figure 4(b). The lines separating the triangles T_j^{int} and T_j^{ext} are denoted E_j^{ext}. Only the part of the control volume composed of the interior triangles T_j^{int} and the fracture sub control-volumes Q_1^f and Q_2^f is considered in the flux recovery, see Figure 4(c).



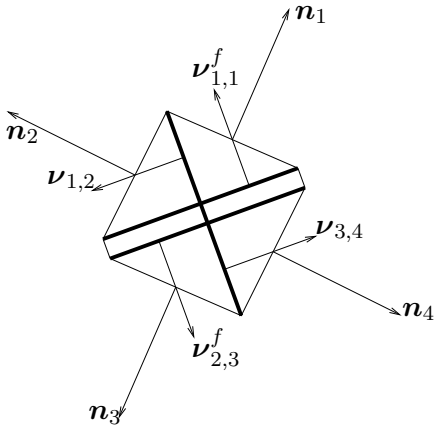
(a) Four elements (triangles) with vertices $V_1, \ldots, V_5$. A control volume (dashed line) is associated with V_5.



(b) The control volume in the left figure. The fracture has been expanded to a 2D element with aperture d.



(c) Control-volume edge numbering



(d) Control-volume normal vectors

FIGURE 4. Flux recovery for a control volume with an internal fracture.

From the flow simulation, fluxes are given over the exterior faces of the control volume Ω_{cv}, as indicated by the dashed line segments in Figure 4(b). The recovery procedure calculates additional conservative fluxes on the interior matrix edges (E_j^{int} and $E_j^{int,f}$) and the interior fracture edge (E^f), see Figure 4(c). These fluxes on E_j^{int} and $E_j^{int,f}$ are obtained indirectly by computing a constant Darcy velocity $\boldsymbol{q}_j$ of each interior triangle T_j^{int}.

The constant Darcy velocities $\boldsymbol{q}_j$ and the fracture interior fluxes F^f must satisfy the following conditions

- mass conservation for the exterior triangle T_j^{ext}:

$$\boldsymbol{q}_j \cdot \boldsymbol{n}_j \;=\; F_{j,1} + F_{j,2}\;, \qquad j = 1, \ldots, N\;, \tag{3}$$

where $\boldsymbol{n}_j$ is the outward normal vector to E_j^{ext} relative to T_j^{ext} with its length equal to the length of the edge E_j^{ext}, see Figure 4(d). Fluxes $F_{j,1}$ and $F_{j,2}$ are the given fluxes with respect to the outward normal vector of T_j^{ext} at the two edges of T_j^{ext} which coincides with the $\partial\Omega_{cv}$, see Figure 4(b).

- the flux over the interior boundaries not coinciding with the fracture edges must be continuous:

$$\boldsymbol{q}_j \cdot \boldsymbol{\nu}_{j,j+1} \;=\; \boldsymbol{q}_{j+1} \cdot \boldsymbol{\nu}_{j,j+1} \;, \qquad j = 1, \ldots, M-1, M+1, \ldots, N-1 \;, \tag{4}$$

  where M is the number of the last interior triangle belonging to Ω_1, $\boldsymbol{\nu}_{j,j+1}$ is the normal vector of the interior boundary E_j^{int} pointing from T_j^{int} to T_{j+1}^{int} and has length equal to E_j^{int}, see Figure 4(d).
- mass conservation in one fracture Q_k^f is required:

$$-\boldsymbol{q}_j \cdot \boldsymbol{\nu}_{k,j}^f + \boldsymbol{q}_{j+1} \cdot \boldsymbol{\nu}_{k,j}^f + F^f + F_k^f \;=\; 0 \;, \tag{5}$$

  where F^f is the unknown flux over the fracture interior edge E^f, F_k^f is the given flux over the edge of the expanded fracture Q_k^f, which coincides with a part of the boundary of the expanded control volume. The sign of the fluxes are chosen according to the outward normal vector of Q_k^f. Further, $\boldsymbol{\nu}_{k,j}^f$ is a normal vector to the edge $E_j^{int,f}$ between the fracture k and the interior triangle T_j^{int} and has its length equal to the edge. The sign of $\boldsymbol{\nu}_{k,j}^f$ is chosen as shown in Figure 4(d). Note that we consider only one mass conservation for one of the fracture quadrilaterals; mass conservation for the other is automatically fulfilled since the sum of the fluxes out of Ω_{cv} is zero.

A system of $2N-1$ linear equations has now been set up, however, a total number of unknown components $2N+1$ ($2N$ from the $\boldsymbol{q}_j$ and 1 from the flux F^f) must be determined. To close the system, we need two more equations, which can be derived by requiring the gradient of the pressure field to be irrotational ([6]). From Equation (2), the Darcy velocity $\boldsymbol{q}$ can be written as

$$\boldsymbol{q} = -\frac{\boldsymbol{K}}{\mu} \nabla p \;. \tag{6}$$

Rearranging Equation (6) and taking the curl of a gradient yield

$$\nabla \times \mu \boldsymbol{K}^{-1} \boldsymbol{q} = -\nabla \times \nabla p \;. \tag{7}$$

Since the curl of a gradient is always zero and the dynamic viscosity μ is constant in this study, we have from Stokes theorem

$$\int_\Omega \nabla \times \boldsymbol{K}^{-1} \boldsymbol{q} d\Omega = \oint_\Gamma \boldsymbol{K}^{-1} \boldsymbol{q} \cdot d\boldsymbol{s} = 0 \;. \tag{8}$$

Here, Ω may be any 2D subdomain of the whole solution domain, and Γ is the 1D boundary of Ω. Equation (8) is now applied over two subdomains, Ω_1 and Ω_2, separated by the fracture.

For this simple case shown in Figure 4(c), the subdomain Ω_1 contains T_1^{int} and T_2^{int} with its boundary Γ_1 corresponding to the counter clockwise sequences of edges E_1^{ext}, E_2^{ext}, $E_2^{int,f}$ and $E_1^{int,f}$. Then, Equation (8) can be written as

$$\oint_{\Gamma_1} \boldsymbol{K}^{-1} \boldsymbol{q} \cdot d\boldsymbol{s} \;=\; \sum_{j=1}^{M} \int_{E_j^{ext}} \boldsymbol{K}_j^{-1} \boldsymbol{q}_j \cdot d\boldsymbol{s} + \int_{E_2^{int,f}} K_F^{-1} \boldsymbol{q}_2^f \cdot d\boldsymbol{s} + \int_{E_1^{int,f}} K_F^{-1} \boldsymbol{q}_1^f \cdot d\boldsymbol{s} \;= 0 \;, \tag{9}$$

where the orientation of integration is counter clockwise, M is the number of the last interior triangle belonging to Ω_1 (here $M = 2$), and the fracture permeability K_F is a scalar.

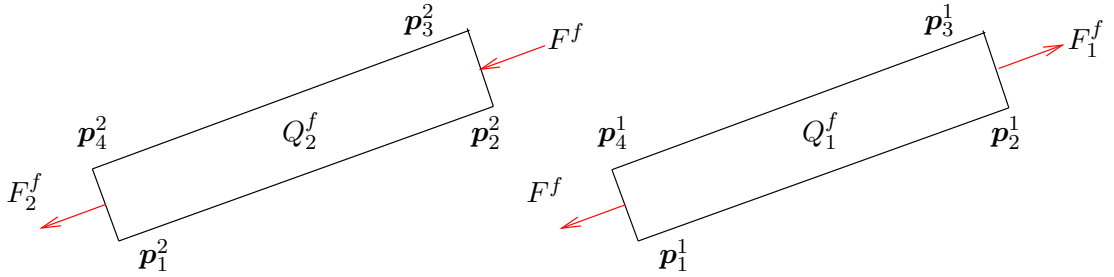


FIGURE 5. The four corners points and the direction of fluxes of the fracture rectangles.

In the first term of Equation (9), both $\boldsymbol{K}_j$ and $\boldsymbol{q}_j$ are constant; hence,

$$\sum_{j=1}^{M} \int_{E_j^{ext}} \boldsymbol{K}_j^{-1}\boldsymbol{q}_j \cdot d\boldsymbol{s} = \sum_{j=1}^{M} \boldsymbol{K}_j^{-1}\boldsymbol{q}_j \cdot \boldsymbol{t}_j = \sum_{j=1}^{M} \boldsymbol{K}_j^{-1}\boldsymbol{t}_j \cdot \boldsymbol{q}_j \ , \tag{10}$$

where the tangent vectors $\boldsymbol{t}_j$ corresponds to a 90 degrees counter-clockwise rotation of the normal vector $\boldsymbol{n}_j$ having the length of E_j^{ext}.
The second and the third terms in Equation (9) are integrals along the fracture edges. The velocities in the fracture quadrilaterals $\boldsymbol{q}_1^f$ and $\boldsymbol{q}_2^f$ are given by linear interpolation of the edge fluxes using Pollock's method [20]. This yields

$$\int_{E_2^{int,f}} K_F^{-1}\boldsymbol{q}_2^f \cdot d\boldsymbol{s} = \frac{(-F_2^f - F^f)\|\boldsymbol{u}_2\|}{2K_F\|\boldsymbol{v}_2\|} \ , \tag{11}$$

and

$$\int_{E_1^{int,f}} K_F^{-1}\boldsymbol{q}_1^f \cdot d\boldsymbol{s} = \frac{(F_1^f - F^f)\|\boldsymbol{u}_1\|}{2K_F\|\boldsymbol{v}_1\|} \ , \tag{12}$$

where

$$\boldsymbol{u}_k = \boldsymbol{p}_2^k - \boldsymbol{p}_1^k \qquad \text{and} \qquad \boldsymbol{v}_k = \boldsymbol{p}_4^k - \boldsymbol{p}_1^k \ , \qquad k = 1, 2 \ . \tag{13}$$

As shown in Figure 5, $\boldsymbol{p}_i^k$ is the coordinate of the corners i of an extended fracture Q_k^f and F_i^f is the flux over the fracture exterior edge E_k^f given from the flow simulation. The details of the calculation leading to Equations (11) and (12) are presented in the Appendix.

Substituting Equations (10) - (12) in Equation (9) yields

$$\oint_{\Gamma_1} \boldsymbol{K}^{-1}\boldsymbol{q} \cdot d\boldsymbol{s} = \sum_{j=1}^{M} \boldsymbol{K}_j^{-1}\boldsymbol{t}_j \cdot \boldsymbol{q}_j + \frac{(-F_2^f - F^f)\|\boldsymbol{u}_2\|}{2K_F\|\boldsymbol{v}_2\|} + \frac{(F_1^f - F^f)\|\boldsymbol{u}_1\|}{2K_F\|\boldsymbol{v}_1\|} \ . \tag{14}$$

A similar argument can be used to show that the line integral along Γ_2 can be given as

$$\oint_{\Gamma_2} \boldsymbol{K}^{-1}\boldsymbol{q} \cdot d\boldsymbol{s} = \sum_{j=M+1}^{N} \boldsymbol{K}_j^{-1}\boldsymbol{t}_j \cdot \boldsymbol{q}_j - \frac{(F_1^f - F^f)\|\boldsymbol{u}_1\|}{2K_F\|\boldsymbol{v}_1\|} - \frac{(-F_2^f - F^f)\|\boldsymbol{u}_2\|}{2K_F\|\boldsymbol{v}_2\|} \ . \tag{15}$$

The general case of a discretized fracture-matrix control volume is described by ϱ fractures meeting at a vertex ($\varrho = 0, 1, 2, ...$). A new mesh of expanded fractures is constructed by introducing a polygon $\mathcal{M}$ with ϱ edges at the overlapping area of the ϱ expanded fractures. The ϱ fractures now becomes ϱ trapezoidal elements and the central polygon $\mathcal{M}$ is divided into ϱ triangles, each having one vertex at the centroid of the polygon. A sketch of a case for $\varrho = 5$ is shown in Figure 6. The flux recovery method for two fractures described

previously can now be extended to the case of ϱ fractures. The exterior flux continuity shown in Equation (3) remains the same. However, the interior continuity equations shown in Equation (4) is reduced from $N-2$ to $N-\varrho$ due to the presence of more fractures within the control volume. Furthermore, there are now ϱ fractures for which Equation (5) has to be imposed. These increased constraints are counterbalanced by an increased number of unknown fracture fluxes F_k^f in Equation (5). Next, Equations (14) and (15) have to be extended to ϱ curl relations, instead of two. Finally, after the system of equations are solved, the solution of the fracture fluxes F_k^f are then used to compute the interior fluxes of the triangles of the central polygon $\mathcal{M}$ by following the original method of Cordes and Kinzelbach [6].

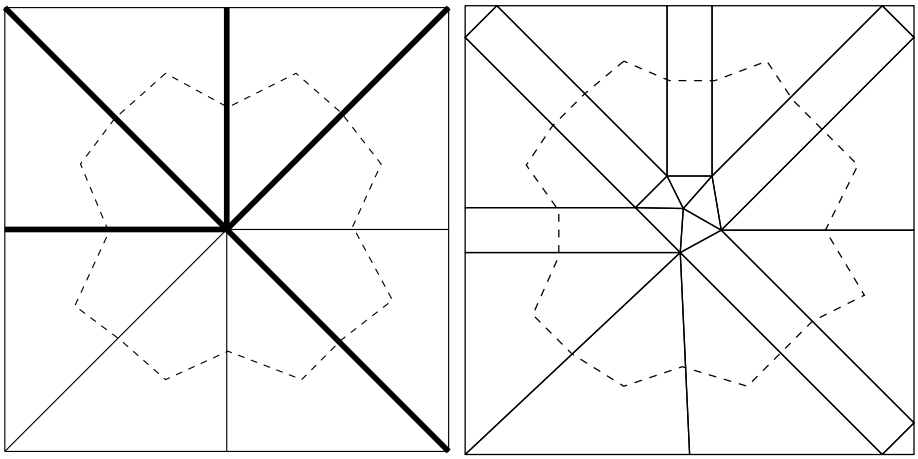


FIGURE 6. Five fracture meeting. Left: Unexpanded fractures. Right: Expanded fractures.

For other types of control volumes, a similar concept of the fracture extension and the flux recovery are applied. The system of equations are solved based on Equations (3), (4), (5), (14) and (15).

This work do not consider the 3D case, however extension of the flux recovery procedure to the 3D case without fractures has been considered by Prévost [21]. A difficulty with the 3D case compared to the 2D case is that in 3D there are more faces connected to a vertex than there are elements. In 2D, the number of edges and the number of elements connected to a vertex were the same, which allowed a straight forward derivation of the linear system. In 3D, additional constraints must be devised to close the system, or the system can be solved in least-square sense, see [21]. The expansion of the fractures for the 3D case is discussed in Hægland [10].

2.3. **Streamline tracing.** A streamline $\boldsymbol{s}(\tau)$ is defined by requiring that the tangent of the streamline should be equal to the velocity,

$$\frac{d\boldsymbol{s}}{d\tau} = \frac{\boldsymbol{q}(\boldsymbol{x})}{\phi} \,, \tag{16}$$

where τ is the streamline parameter denoted the time-of-flight (TOF), $\boldsymbol{q}$ is the Darcy velocity, and ϕ is the porosity. By rearranging Equation (16) and integrating with respect to the arc length of a streamline, the TOF that a particle need to travel a given distance s is given by,

$$\tau(s) = \int_0^s \frac{\phi}{\|\boldsymbol{q}\|} ds', \tag{17}$$

where s measures arc length along a streamline. Note that, due to the appearance of the porosity in Equation (17), the TOF is related to the particle velocity, not the Darcy velocity.

Methods for streamline tracing on quadrilateral grids when fluxes are known have been investigated by several authors. For a regular quadrilateral mesh (rectangular mesh) Pollock's method has been widely used. The method assumes a piece-wise linear approximation of the velocity over the entire grid. Within a single grid cell taken to be the unit square for simplicity, the velocity is given as

$$\boldsymbol{q}(\boldsymbol{x}) = \begin{bmatrix} f_{x0}(1-x) + f_{x1}x \\ f_{y0}(1-y) + f_{y1}y \end{bmatrix}, \qquad 0 \le x \le 1 \quad \text{and} \quad 0 \le y \le 1\,, \tag{18}$$

where f_k are fluxes over the cell faces (see Figure 7). Solving Equation (16) by inserting the velocity from Equation (18) yields two separate expressions for the TOF:

$$\tau_x(x_i, x_j) = \frac{\phi}{f_{x1} - f_{x0}} \ln\left(\frac{f_{x0} + (f_{x1} - f_{x0})x_j}{f_{x0} + (f_{x1} - f_{x0})x_i}\right), \tag{19}$$

and

$$\tau_y(y_i, y_j) = \frac{\phi}{f_{y1} - f_{y0}} \ln\left(\frac{f_{y0} + (f_{y1} - f_{y0})y_j}{f_{y0} + (f_{y1} - f_{y0})y_i}\right). \tag{20}$$

The TOF that a particle travels from the entry point $\boldsymbol{x}_{en}$ to the exit point $\boldsymbol{x}_{ex}$ of the grid cell is determined by calculating the time that a streamline requires to cross the grid boundaries. Inserting $x_j = 0$ and 1 in Equation (19) and $y_j = 0$ and 1 in Equation (20), and replacing x_i and y_i with $\boldsymbol{x}_{en}$ yield four different times that the streamline requires to cross the left, the right, the bottom and the top boundaries respectively. The TOF is the minimum positive time of the calculated times. By rearranging Equations (19) and (20) and inserting the TOF in τ_{ex}, the exit point $\boldsymbol{x}_{ex}$ is then given as

$$x_{ex} = \frac{1}{f_{x1} - f_{x0}} \left\{ q_{en,x} \exp\left(\frac{\tau_{ex}}{\phi}(f_{x1} - f_{x0})\right) - f_{x0} \right\}, \tag{21}$$

$$y_{ex} = \frac{1}{f_{y1} - f_{y0}} \left\{ q_{en,y} \exp\left(\frac{\tau_{ex}}{\phi}(f_{y1} - f_{y0})\right) - f_{y0} \right\}, \tag{22}$$

where $\boldsymbol{q}_{en}$ is the velocity at the entry point $\boldsymbol{x}_{en}$ calculated from Equation (18).

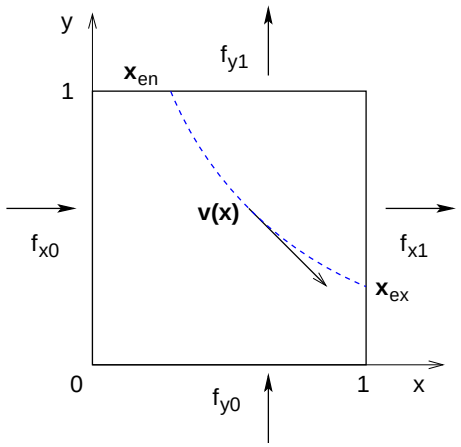
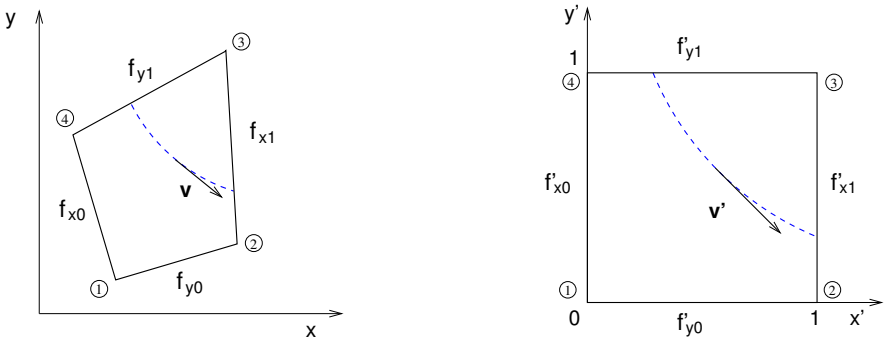


FIGURE 7. Pollock tracing for a unit square.

A complex fracture-matrix system can only be discretized precisely with unstructured grids. Hence, streamline tracing which performs well on unstructured grids is required. The extension of Pollock's method on unstructured grids has proven successful in several studies [6, 22, 11]. The spatial coordinates together with the velocity in the physical space $\mathcal{P}$ are transformed to a reference space $\mathcal{R}$ by using the bilinear iso-parametric transformation, see Figure 8.



(a) Physical space $\mathcal{P}$ (b) Reference space $\mathcal{R}$

FIGURE 8. Transformation of an unstructured grid and edge fluxes from a physical space $\mathcal{P}$ to a reference space $\mathcal{R}$.

According to Hægland et al. [11], the velocity field $\boldsymbol{v}'$ in $\mathcal{R}$ is related to the linear flux interpolation as

$$\boldsymbol{q}' = \frac{d\boldsymbol{x}'}{dt} = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} f_{x0}(1-x'_i) + f_{x1}x'_i \\ f_{y0}(1-y'_i) + f_{y1}y'_i \end{bmatrix} , \tag{23}$$

where $\boldsymbol{J}$ is the Jacobian transformation matrix

$$\boldsymbol{J} = \begin{bmatrix} \dfrac{dx}{dx'} & \dfrac{dx}{dy'} \\ \dfrac{dy}{dx'} & \dfrac{dy}{dy'} \end{bmatrix} . \tag{24}$$

The velocity in Equation (23) is rewritten in terms of a pseudo time τ in $\mathcal{R}$ as shown by Jimenez et al. [13] as

$$d\tau = \frac{dt}{\det \boldsymbol{J}} = \begin{bmatrix} \dfrac{dx'}{f_{x0}(1-x'_i) + f_{x1}x'_i} \\ \dfrac{dy'}{f_{y0}(1-y'_i) + f_{y1}y'_i} \end{bmatrix} , \tag{25}$$

where t is real time in $\mathcal{P}$. The actual time-of-flight t_{ex} is then evaluated by integrating Equation (25) from $\boldsymbol{x}'_{en}$ to $\boldsymbol{x}'_{ex}$:

$$t_{ex} = \int_0^{t(\tau_{ex})} dt = \int_0^{\tau_{ex}} \det \boldsymbol{J}(x'(\tau), y'(\tau)) d\tau . \tag{26}$$

Recently, some problems with the method have been reported and resolved. Inaccuracies in computing TOF due to errors in the absolute value of the interpolated velocity field have been reported in [13, 9, 11]. Jimenez et al. [13] proposed an extension of the method that allowed for exact reproduction of time-of-flight for uniform flow in 2D. In this paper, we utilize this latter approach, see [9, 11, 13] for more details of the method.

2.4. **Evaluation of the breakthrough curve.** We assume purely advective transport of a solute tracer in the streamline method and visualize each streamline as a flow channel or a streamtube. Due to the pure advective transport in the streamline method, no mass exchanges between neighboring streamtubes. The TOF measures the time that the tracer needs to travel along the streamtube, see Figure 9. Streamlines are distributed equally according to the total flux along the inflow boundary, such that each streamtube contains the same flux.

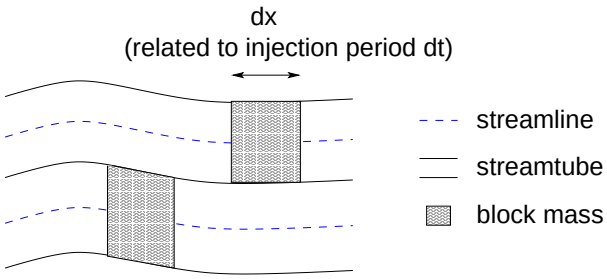


FIGURE 9. Mass transport in a streamtube as a block.

The transport behavior characterized from the results of the streamline method are described using a breakthrough curve (BTC) and an accumulated breakthrough curve (AccBTC). The latter is the sum of the total mass leaving the domain at the outflow boundary until a time t and BTC is the rate of change of AccBTC during the time interval Δt

$$\mathrm{BTC}(t) = \frac{\mathrm{AccBTC}(t) - \mathrm{AccBTC}(t-\Delta t)}{\Delta t}\,, \tag{27}$$

$$\mathrm{AccBTC}(t) = \mathrm{AccBTC}(t-\Delta t) + \dot{m}^s \sum_{i=1}^{n} \mathcal{T}_i\,, \tag{28}$$

where $\dot{m}^s$ is the normalized mass flux, which is determined by the mass flux in each streamtube over the total injected mass, and n is the total number of the streamtubes. The arrival time condition $\mathcal{T}_i$, for a streamtube i, is given as

$$\mathcal{T}_i(t) = \begin{cases} 0 & ; \quad t < \mathrm{TOF}_i \\ t - \mathrm{TOF}_i & ; \quad \mathrm{TOF}_i \le t \le \mathrm{TOF}_i + dt \\ 0 & ; \quad t > \mathrm{TOF}_i + dt \end{cases} \tag{29}$$

where the time of flight TOF_i is the time that a block mass in a streamtube i travels until it reaches the outflow boundary and dt is the duration of mass injection.
The TOF of each streamtube is a discrete value, which can be the same for all streamtubes or highly varied depending on the geometries and structures of a domain. The overall transport behavior of the system is presented as a histogram BTC which evaluates the rate of change of the normalized mass flux over a specified interval of time.

3. Comparison Results

3.1. **Preliminary test case.** In order to understand the difference arising from simulating the transport process using grid-based advective transport (ADT) and the streamline method (STR), three preliminary test cases are set up: a single short fracture, a single long fracture and systematically distributed fractures, see Figure 10 (top). For all test cases, a two-dimensional domain of 1.0×1.0 m is set up. The fluid properties and the domain properties correspond to the data presented in Table 1.

We consider transport of a solute tracer without dispersion and the governing equation can be written as

$$\frac{\partial c}{\partial t} + \nabla \cdot \frac{\boldsymbol{q}}{\phi} c = 0 \,. \tag{30}$$

where c is the concentration of the tracer. For the ADT, Equation (30) is discretized using a box method with upwinding formulation ([19]). For the STR, the transport of solute tracer is obtained directly from the streamline time-of-flight and Equation (28).
The boundary conditions are no-flow on the top and the bottom. For the flow simulation, a Dirichlet boundary in terms of pressure is given on the left-hand side (the inflow boundary) and on the right-hand side (the outflow boundary). For the grid-based transport simulation (ADT), tracer is given for a very short time at the inflow boundary. A free-flow boundary is given at the outflow for the transport simulation, which means that a tracer arriving at the boundary is allowed to leave freely. A total of 500 streamlines are traced and distributed equally according to the fluxes at the inflow. We used 500 streamlines because the AccBTC from 500 streamlines showed visual convergence to the AccBTC from 10000 streamlines.
The ADT and the STR are carried out on different mesh sizes (measured in terms of the average length of the sides of a grid cell) of approximately 0.01, 0.02 and 0.03 m to investigate the influence of numerical diffusion.

		Domain Properties	
		matrix	**fracture**
Permeability, K	$[m^2]$	$1.0 \cdot 10^{-13}$	$8.33 \cdot 10^{-10}$
Eff. porosity, ϕ	[-]	0.13	0.30
Aperture, b	$[m]$	-	$1.0 \cdot 10^{-4}$
		Fluid Properties	
Viscosity, μ	$[kg/(m \cdot s)]$	$1.814 \cdot 10^{-5}$	
Density, ρ	$[kg/m^3]$	1.21	

TABLE 1. Domain and fluid properties for all simulations.

Some streamlines of the fine mesh (0.01 m) are shown in Figure 10 (bottom). In all cases we observed that high permeability in fractures resulted in flow paths towards the fractures. These flow paths yield a faster tracer transport within their streamtubes, whereas the other flow paths in the surrounding matrix lead to a slower tracer transport.
The results of the accumulated breakthrough curves (AccBTCs) and the breakthrough curves (BTCs) are presented in Figure 11. In all test cases, the AccBTCs and the BTCs of the ADT differ considerably between different mesh sizes due to numerical diffusion. Whereas, the AccBTCs from the STR are rather similar for all mesh sizes and their slight local differences come from the fact that the velocity field obtained from the flow simulation is locally grid dependent, especially near the fractures.
In the short fracture case, the BTCs of the ADT in Figures 11a1 show that using a coarser mesh size results in a higher variance and a lower peak value. This indicates that the numerical diffusion increases the spreading of the tracer plume in the matrix and smears out the concentration front, especially the double-continuum effect due to fast transport in the fracture and slow transport in the matrix has disappeared. On the contrary, for the STR having no numerical diffusion, the double-continuum effect results in the two jumps of the AccBTCs (Figure 11a2) and the two peaks of the BTCs (Figure 11a3). The large

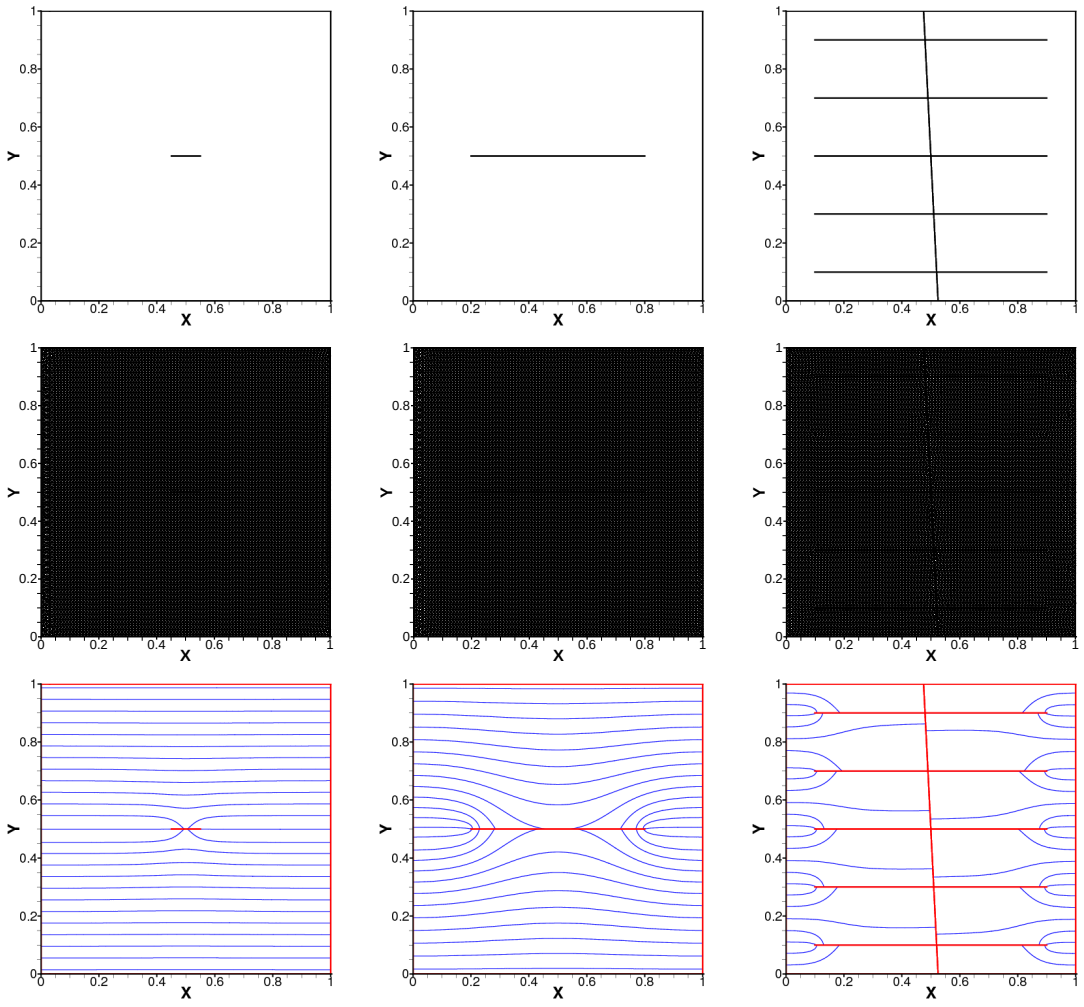


FIGURE 10. Top: Domains with fractures. Middle: Discretization domains (top) for a mesh size of 0.01 m corresponding to ca. 11700 vertices. Bottom: 25 of the 500 streamlines traced for the grids in the middle row.

part of the tracer transported in the matrix arrives about the same time at the outflow and results in a very high mass flux in the second peak of the BTC and a sharp rise of the second jump in the AccBTC.

When the fracture becomes longer (Figure 11b), the double-continuum effect can also be noticed for the ADT, in spite of the numerical diffusion, see Figures 11b1 and 11b2. The fast transport in the long fracture results in the first peak of the BTCs; later, the part of the tracer plume transported in the matrix leads to the second peak. The numerical diffusion in the transversal flow-direction causes spreading of mass transported in the fracture to the surrounding matrix. Therefore, the value of the first peak of the BTC of the STR is higher than that of the ADT (see Figure 11b3) and the AccBTCs show a sharp rise for the STR, but only a gradual increase for the ADT (see Figures 11b2). This effect delays the arrival time of the mass transported in the fracture. The influence of the numerical diffusion on the part of the tracer transported through the porous matrix shows the same behavior as that observed in the BTCs for the single short fracture, as discussed above.

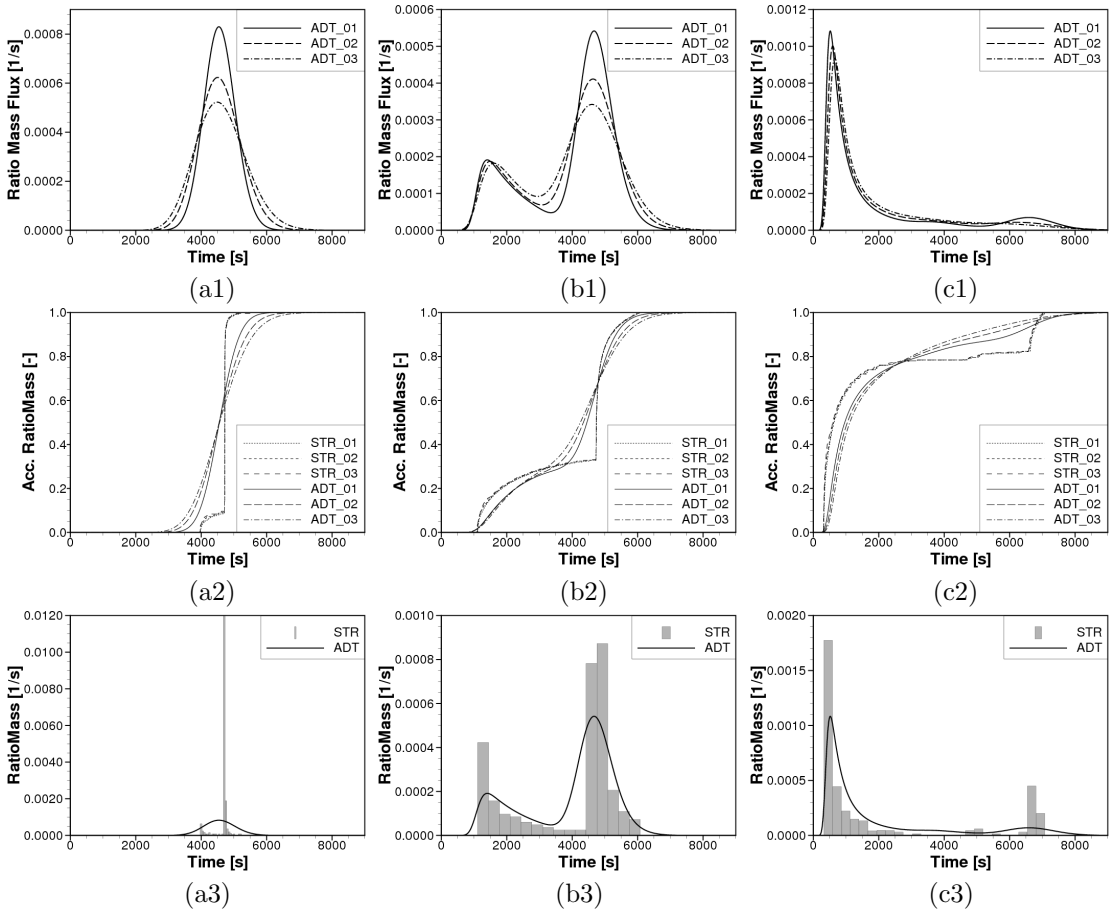


FIGURE 11. Transport simulation results of ADT and STR for the test cases: a) (left column) a single short fracture, b) (middle column) a single long fracture and c) (right column) systematically distributed fractures. Top row (a1/b1/c1): BTC of ADT for different mesh sizes. Middle row (a2/b2/c2): AccBTC of ADT and STR for different mesh sizes. Bottom row (a3/b3/c3): BTC of ADT and STR at a mesh size of 0.01 m.

Increasing the number of horizontal fractures with a vertical fracture connecting all horizontal fractures (Figure 11c) increases the part of tracer transported in the fractures and decreases the part transported in the matrix. Therefore, the BTCs in Figure 11c1 show a high peak and the AccBTCs in Figure 11c2 show a high first jump. Due to the influence of the numerical diffusion in the ADT, the BTCs of the coarse mesh size of 0.03 and 0.02 m show only a long tail, whereas the BTC of the fine mesh size of 0.01 m has a small second peak (see Figure 11c1), more similar to the BTC of the STR showing a double-continuum effect as can be seen in Figure 11c3. The effect of the numerical diffusion along fractures in the transversal flow-direction can be better noticed in this case than in the single long fracture case. As the mesh gets finer and the numerical diffusion decreases, the tracer transported in the fractures remains more confined to the fractures and arrives faster at the outflow and this yields a BTC with a slightly higher peak concentration shifted somewhat to the left compared to the BTCs of the coarser mesh sizes (see Figure 11c1). The AccBTCs of the ADT seem to converge to the result of the STR when the mesh becomes finer and the numerical diffusion decreases, see Figure 11c2.

3.2. **A complex fracture-matrix system.** After a basic understanding of the transport behavior for ADT and STR has been gained from the preliminary test cases, the next step is to perform a comparison study between the two approaches in a complex fracture-matrix system. The fracture networks are generated by the fracture generator FRAC3D based on statistical geometries and geostatistical parameters taken from a study site in Pliezhausen, Germany, see [2, 3]. Boundary conditions, fluid properties, and domain properties are the same as for the test cases in the previous section.
We consider five different cases (GFG-A, GFG-B, GFG-C, SFG-A and SFG-B) where their fracture networks are generated from five different geometries and spatial characteristics using a statistical or geostatistical approach as described in the associated paper ([3]). Several realizations of a fracture network with the same fracture geometries and spatial characteristics can be obtained when a fracture network is generated based on a stochastical approach. In this study, for each of the five cases we consider five fracture-network realizations. We first consider the five realizations from the GFG-A, and subsequently consider the average behavior of the five realizations from each of the cases.
We start by generating five realizations R1-R5 from the GFG-A. The streamline distribution taken from realization R1 is shown in Figure 12. In a complex fracture-matrix system, the increased number of fractures and varying orientations lead to a high dispersion of the tracer plume. As shown in Figure 12, no streamlines are transported only in the surrounding matrix, but rather they are partly transported in the fractures and partly in the matrix. Hence, the AccBTCs of the five realizations do not show a clear double-continuum behavior, see Figure 13. For all realizations, the AccBTCs of the STR shift to the left compared with the ADT which means that in the STR simulation mass transports faster than in the ADT simulation. This corresponds with the results presented in the case of systematically distributed fractures. The numerical diffusion in the ADT leads to a delay of mass transport in the fractures.

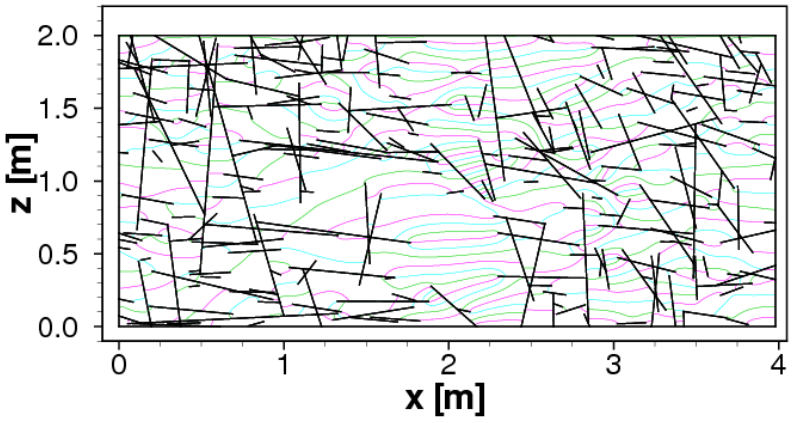
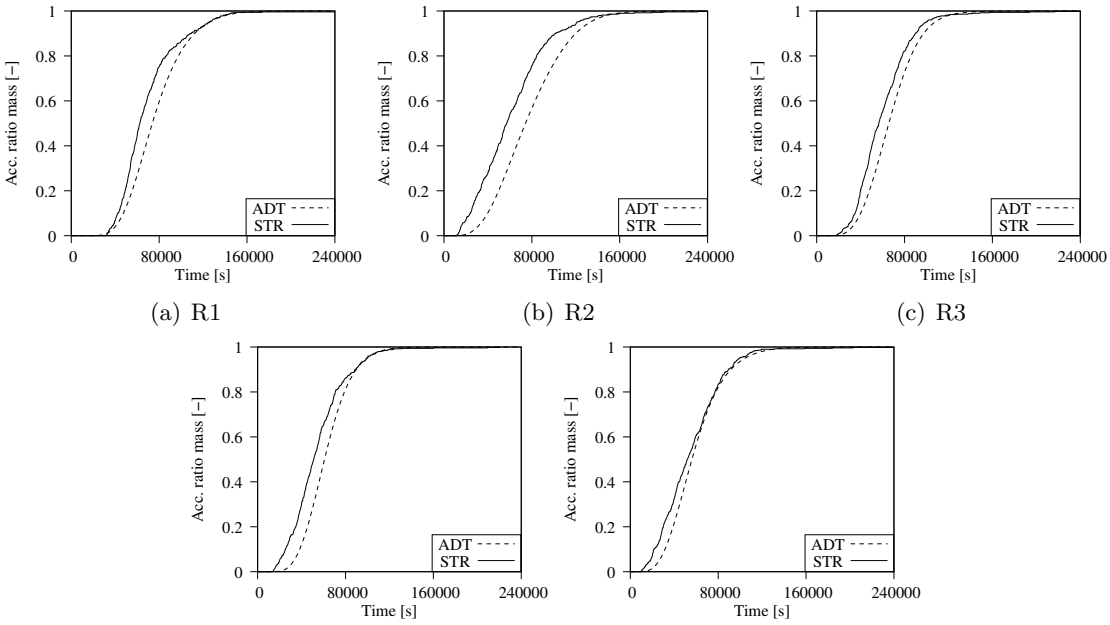


FIGURE 12. Streamlines traced for one realization (R1) of a complex fracture-matrix system of the GFG-A.

The transport behavior is characterized by the effective travel time, the variance and the skewness and they are evaluated from the moment and the central moment of the BTC ([3]). The variance and the skewness of the BTC depend highly on the distribution of the tracer concentration. A high variance means a high degree of spreading of the plume. The skewness represents the asymmetry of the spreading. A positive skewness indicates that a larger part of tracer plume first transports to the outflow, and then the rest of the

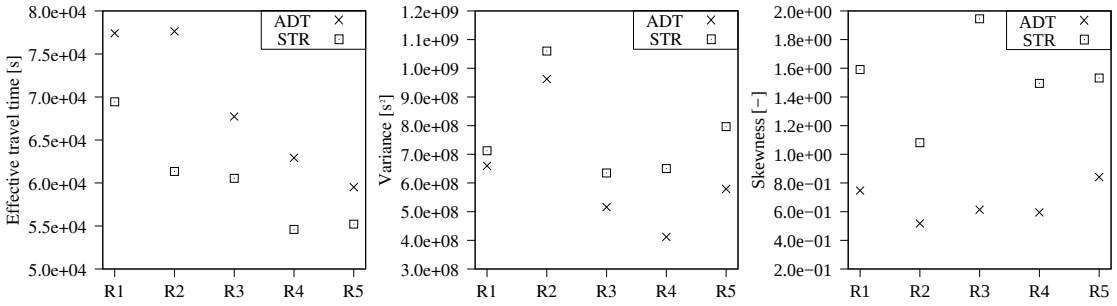


(a) R1 (b) R2 (c) R3

(d) R4 (e) R5

FIGURE 13. AccBTCs of ADT and STR for the complex fracture-matrix systems for five realizations (R1-R5) of the GFG-A.

plume arrives gradually. The faster mass transport in the STR leads to a less effective travel time compared with the transport in the ADT (see Figure 14a). For all realizations, higher variances and higher skewnesses are seen in the STR compared to the ADT (see Figures 14b and 14c). This is due to the fact that the fast transport in the fractures and the slow transport in the matrix are better captured with the STR, whereas the numerical diffusion in the ADT smears out this contrast.

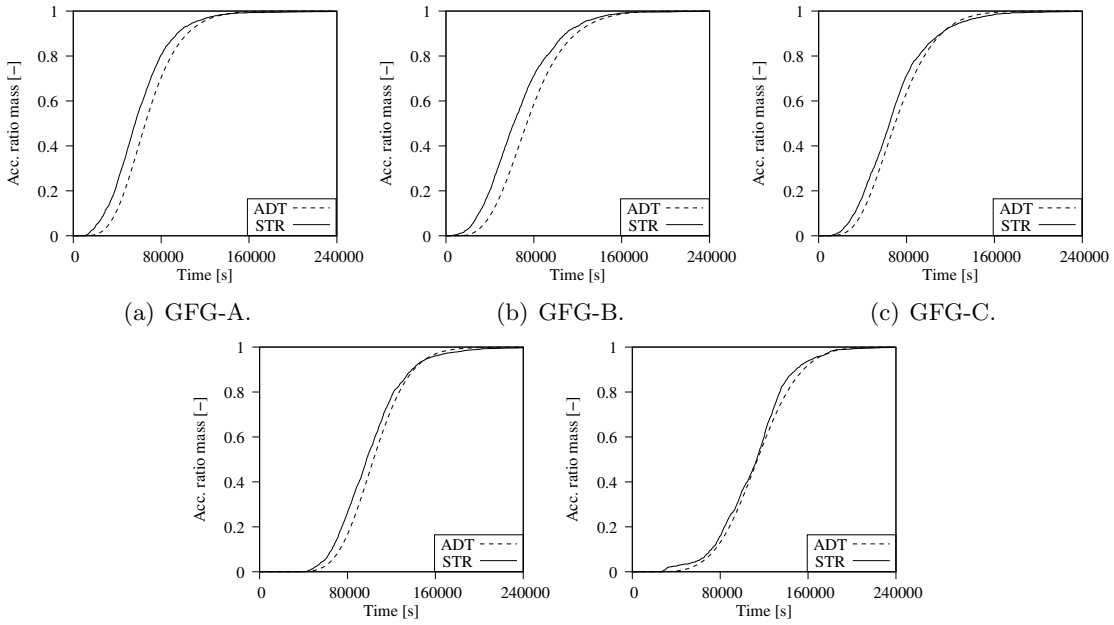


(a) Effective travel time. (b) Variance. (c) Skewness.

FIGURE 14. Quantitative results of ADT and STR for the complex fracture-matrix systems of five realizations from the GFG-A.

In the last part of this section we consider the average behavior for the five cases GFG-A, GFG-B, GFG-C, SFG-A and SFG-B. The details about the fracture generation are presented in the associated paper ([3]). For each case, five realizations are generated and the results presented for each case are the average values over all five realizations.

The average AccBTCs from the STR and the ADT for the five cases are shown in Figure 15. For all five study cases, the average AccBTCs from the STR shifts to the left compared with the ADT which means that the STR leads to a faster mass transport compared to the ADT. This agrees with the average values of the effective travel times of the STR being less than those of the ADT, as shown in Figure 16a. In all cases, a physical dispersion due to fast and slow transport can be identified from the positive skewness in Figure 16c. Due to the purely advective transport and no numerical diffusion in the STR, the variance and the skewness for the STR are larger than for the ADT, indicating a larger physical dispersion. On the other hand, the numerical diffusion in the ADT delays plume migration in the fracture-matrix system. As a result, the difference between AccBTCs and BTCs of the two approaches is clearly noticed for all study cases of the fracture-matrix system. Most of the values obtained from the STR show larger extreme values than from the ADT, particularly for the variance of the SFG-B. Hence, more realizations might be required to represent the transport behavior using STR.



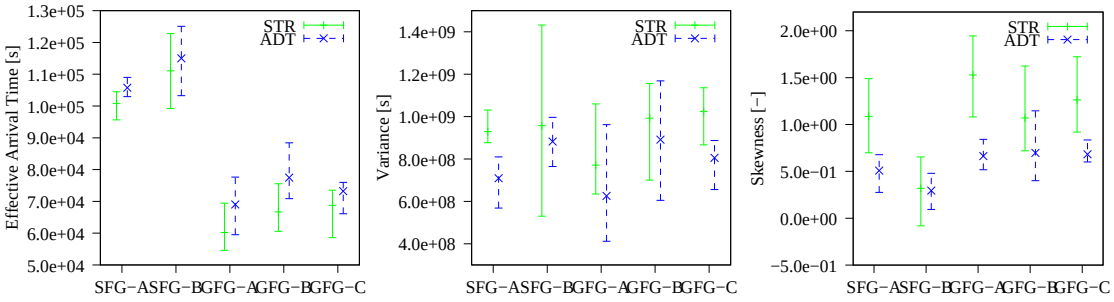
(a) GFG-A. (b) GFG-B. (c) GFG-C.

(d) SFG-A. (e) SFG-B.

FIGURE 15. Average AccBTCs of ADT and STR for the complex fracture-matrix systems for the five different cases GFG-A, GFG-B, GFG-C, SFG-A and SFG-B.

4. Conclusion

The precision of streamline tracing strongly depends on the accuracy of the velocity field. Approximating the fluxes from pressure gradients obtained from the flow simulation results in discontinuous fluxes at element boundaries. A flux recovery for a two-dimensional fracture-matrix system is proposed for calculating mass conservative fluxes over boundaries of quadrilateral sub control-volumes. Lower-dimensional fractures, which are assumed in the flow simulation, have to be extended to equi-dimensional fractures to obtain well defined velocities and streamlines in the fractures (not only parallel to the orientation of the fractures).



(a) Effective travel time. (b) Variance. (c) Skewness.

FIGURE 16. Average and extreme values (min./max.) obtained from the ADT and the STR for the complex fracture-matrix systems for the five different cases.

The applicability of a streamline method (STR) for the study of the transport behavior in a fracture-matrix system is investigated by comparing with the results from a grid-based advective transport (ADT) model. In the simple cases of one fracture or systematically distributed fractures, the effect of fast flow in the fractures and slow flow in the matrix is smeared out due to the numerical diffusion in the ADT. The preferential flow paths in the fracture-matrix system are clearly noticed in the STR from the double-peak BTCs and two sudden rises in the AccBTCs. In the complex fracture-matrix system consisting of a large number of fractures with varying orientations, numerical diffusion in the ADT delays plume migration, whereas purely advective transport in the STR leads to fast solute transport and maintains a high physical dispersion due to the fast transport in the fractures and slow transport in the matrix. As a result, we observe a less effective travel time, higher variance and higher skewness from the STR than from the ADT as well as a shift of AccBTC of the STR to the left.
Further investigations involving comparisons with experimental or field studies have to be carried out in order to validate the results of the STR. If the assumption of the purely advective transport in the STR leads to an acceleration of the tracer transport in the system compared with the measurement results, including dispersive transport in the STR could be considered to improve the STR approach.

5. APPENDIX

The integral of the velocity along the fracture edge is computed by assuming that the velocities in the quadrilateral fracture are given by linear interpolation of the edge fluxes similar to Pollock's method ([20]). Following Prévost et al. [22] and Jimenez et al. [13], the rectangle Q_1^f (see Figure 4(b)) in $\mathcal{P}$ is transformed to a unit square in a reference space $\mathcal{R}$ using the bilinear transformation $\boldsymbol{x}(\hat{x}, \hat{y})$, which simplifies for a rectangle (Figure 17 (right)) to,

$$\boldsymbol{p}(\hat{x}, \hat{y}) = \boldsymbol{p}_1 + (\boldsymbol{p}_2 - \boldsymbol{p}_1)\hat{x} + (\boldsymbol{p}_4 - \boldsymbol{p}_1)\hat{y} \,, \tag{31}$$

with the constant Jacobian matrix,

$$\boldsymbol{J} = \begin{bmatrix} x_2 - x_1 & x_4 - x_1 \\ y_2 - y_1 & y_4 - y_1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} & \boldsymbol{v} \end{bmatrix}, \tag{32}$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are the shape vectors of the rectangle, and $\boldsymbol{x}_i$ is the point at corner i see Figure 17 (left). Edge fluxes F_{x0}, F_{x1}, F_{y0}, and F_{y1}, are defined for Q_1^f with positive

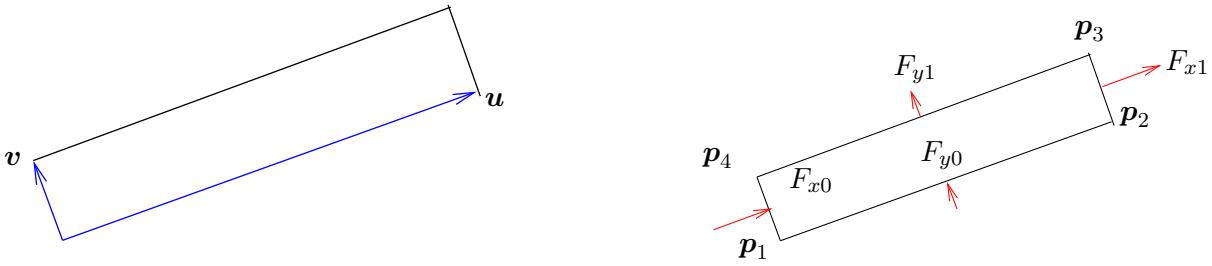


FIGURE 17. The extended fracture rectangle. Left: Shape vectors. Right: Corners and direction of fluxes.

direction as shown in Figure 5 (right). Then, the interpolated velocity $\boldsymbol{q}^f$ in Q_1^f is defined in $\mathcal{R}$ by ([22])

$$\hat{\boldsymbol{q}}^f = \frac{1}{\det \boldsymbol{J}} \begin{bmatrix} F_{x0}(1-\hat{x}) + F_{x1}\hat{x} \\ F_{y0}(1-\hat{x}) + F_{y1}\hat{x} \end{bmatrix}, \tag{33}$$

where the determinant of the Jacobian, $\det \boldsymbol{J}$, is simply equal to the area A of the rectangle. The fluxes F_{y0} and F_{y1} over the edges for the expanded fracture segment Q_1^f are given by the dot product of the constant velocity in the neighboring interior triangle of the fracture and the normal vector of the edge. The sign of the fluxes are indicated in Figure 17 (right). The flux F_{x1} and F_{x0} are given by the unknown interior fracture flux F^f and the flux at fracture flux point respectively. For the expanded fracture segment Q_1^f in Section 2.2, $F_{x0} = -F^f$, $F_{x1} = F_1$, $F_{y0} = -\boldsymbol{q}_N \cdot \boldsymbol{\nu}_{1,1}^f$, and $F_{y1} = -\boldsymbol{q}_1 \cdot \boldsymbol{\nu}_{1,1}^f$. The relation between the velocity in $\mathcal{P}$ and $\mathcal{R}$ is ([11])

$$\boldsymbol{q}^f \;=\; \boldsymbol{J}\hat{\boldsymbol{q}}^f \;=\; \frac{1}{A_1}\left[(F_{x0}(1-\hat{x}) + F_{x1}\hat{x})\boldsymbol{u} + (F_{y0}(1-\hat{y}) + F_{y1}\hat{y})\boldsymbol{v}\right] .$$

The last term in Equation (8) can now be evaluated. We parameterize the path along the fracture edge from $\boldsymbol{p}_4$ to $\boldsymbol{p}_3$ (Figure 17 (right)) as

$$\boldsymbol{\alpha}(\hat{x}) = \boldsymbol{p}(\hat{x}, 1), \qquad 0 \le \hat{x} \le 1 . \tag{34}$$

Then $\boldsymbol{\alpha}'(\hat{x}) = \boldsymbol{u}_1$, and using that $\boldsymbol{u}_1 \cdot \boldsymbol{v}_1 = 0$, $\boldsymbol{u}_1 \cdot \boldsymbol{u}_1 = \|\boldsymbol{u}_1\|^2$, and $A_1 = \|\boldsymbol{u}_1\|\|\boldsymbol{v}_1\|$, we have

$$\begin{aligned} \int_{E_1^{int,f}} K_F^{-1}\boldsymbol{q}_1^f \cdot d\boldsymbol{s} \;&=\; \int_0^1 K_F^{-1}\boldsymbol{q}_1^f(\boldsymbol{\alpha}(\hat{x})) \cdot \boldsymbol{\alpha}'(\hat{x}) d\hat{x} \\ &=\; \frac{1}{A_1}\int_0^1 (F_{x0}(1-\hat{x}) + F_{x1}\hat{x}) K_F^{-1}\boldsymbol{u}_1 \cdot \boldsymbol{u}_1 d\hat{x} + \frac{F_{y1}}{A_1}\int_0^1 K_F^{-1}\boldsymbol{v}_1 \cdot \boldsymbol{u}_1 d\hat{x} \\ &=\; \frac{(F_{x0}+F_{x1})\|\boldsymbol{u}_1\|}{2K_F\|\boldsymbol{v}_1\|} . \end{aligned} \tag{35}$$

6. ACKNOWLEDGMENTS

This work was supported by the International Research Training Group "Non-Linearities and Upscaling in Porous Media (NUPUS)" financed by the German Research Foundation (DFG) and in parts by the Norwegian Research Council, StatoilHydro, and Norske Shell under grant no. 178013/13. The authors would like to thank Jan Nordbotten for useful suggestions and discussion regarding the flux recovery procedure.

References

[1] Al-Huthali, A., and A. Datta-Gupta, Streamline simulation of counter-current imbibition in naturally fractured reservoir, *J. Petrol. Sci. Eng., 43(3-4)*, 271-300, 2004.

[2] Assteerawatt, A., *Flow and Transport Modelling of Fractured Aquifers based on a Geostatistical Approach*, (http://elib.uni-stuttgart.de/opus/volltexte/2008/3639/), PhD Thesis, Universität Stuttgart, Institute of Hydraulic Engineering, 2008.

[3] Assteerawatt, A., H. Hægland, R. Helmig, A. Bárdossy and H. K. Dahle, Simulation of flow and transport processes in a discrete fracture-matrix system I. geostatistical generation of fractures on an aquifer analogue scale, *Water Resour. Res.*, submitted.

[4] Bear, J., *Dynamics of Fluids in Porous Media*, Academic Press, California, 1972.

[5] Cockburn, B., J. Gopalakrishnan, and H. Wang, Locally conservative fluxes for the continuous Galerkin method, *SIAM J. Numer. Anal., 45(4)*, 1742-1776, 2007.

[6] Cordes, C., and W. Kinzelbach, Continuous groundwater velocity fields and path lines in linear, bilinear, and trilinear finite elements, *Water Resour. Res., 28(11)*, 2903-2911, 1992.

[7] Correa, M.R., and A.F.D. Loula., Stabilized velocity post-processing for Darcy flow in heterogeneous porous media, *Commun. Numer. Meth. En., 23*, 461-489, 2007.

[8] Durlofsky, L.J., Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities, *Water Resour. Res., 30(4)*, 965-973, 1994.

[9] Hægland, H., Streamline tracing on irregular grids (http://www.mi.uib.no/~hakonh/thesis/thesis.pdf), *Master's thesis, University of Bergen, Dept. of Mathematics*, 2003.

[10] Hægland, H., A. Assteerawatt, H. Dahle, G.T. Eigestad, and R. Helmig, Comparison of cell- and vertex centered discretization methods for flow in a discrete fracture-matrix system, in preparation.

[11] Hægland, H., H.K. Dahle, G.T. Eigestad, K.-A. Lie, and I. Aavatsmark, Improved streamlines and time-of-flight for streamline simulation on irregular grids, *Adv. Water Resour., 30(4)*, 1027-1045, 2007.

[12] Huang, W., G.D. Donato, and M.J. Blunt, Comparison of streamline-based and grid-based dual porosity simulation, *J. Petrol. Sci. Eng., 43(1-2)*, 129-137, 2004.

[13] Jimenez, E., K. Sabir, A. Datta-Gupta, and M.J. King, Spatial error and convergence in streamline simulation, *SPE Reserv. Eval. Eng., 10(3)*, 221-232, June 2007.

[14] Karimi-Fard, M., L.J. Durlofsky, and K. Aziz, An efficient discrete fracture model applicable for general purpose reservoir simulators, *SPE J.*, 227-236, June 2004.

[15] King, M.J., and A. Datta-Gupta, Streamline simulation: A current perspective, *In Situ, 22(1)*, 91-140, 1998.

[16] Lee, S.H., M.F. Lough, and C.L. Jensen Hierarchical modeling of flow in naturally fractured formations with multiple length scales, *Water Resour. Res., 37(3)*, 443-455, 2001.

[17] Ma, J., G.D. Couples, and S.D. Harris, A mixed finite element technique based on implicit discretization of faults for permeability upscaling in fault damage zones, *Water Resour. Res., 42*, W08413, doi:10.1029/2005WR004686, 2006.

[18] Matringe, S.F., R. Juanes and H.A. Tchelepi, Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids, *J. Comput. Phys., 219(2)*, doi:10.1016/j.jcp.2006.07.004, 2006.

[19] Neunhäuserer, L., *Diskretisierungsansätze zur Modellierung von Strömungs- und Transportprozessen in geklüftet-porösen Medien* (http://elib.uni-stuttgart.de/opus/volltexte/2003/1477/), PhD Thesis, Universität Stuttgart, Institute of Hydraulic Engineering, 2003.

[20] Pollock, D.W., Semi-analytical computation of path lines for finite-difference models, *Ground Water, 26(6)*, 743-750, 1988.

[21] Prévost, M. *Accurate coarse reservoir modeling using unstructured grids, flow-based upscaling and streamline simulation* (http://geothermal.standford.edu/pereposts/search.htm), PhD Thesis, University of Standford, 2003.

[22] Prévost, M., M.G. Edwards, and M.J. Blunt, Streamline tracing on curvilinear structured and unstructured grids, *SPE J.*, 139-148, June 2002.

[23] Reichenberger, V., H. Jakobs, P. Bastian, and R. Helmig, A mixed-dimensional finite volume method for two-phase flow in fractured porous media, *Adv. Water Resour., 29(7)*, 1020-1036, 2006.

[24] Silberhorn-Hemminger, A., *Modellierung von Kluftaquifersystemen: Geostatistische analyse und deterministisch – stochastische Kluftgenerierung* (http://elib.uni-stuttgart.de/opus/volltexte/2003/1278/), PhD Thesis, Universität Stuttgart, Institute of Hydraulic Engineering, 2002.

[25] Srivastava, R., and M.L. Brusseau, Darcy velocity computations in the finite element method for multidimensional randomly heterogeneous porous media, *Adv. Water Resour., 18(4)*, 191-201, 1995.
[26] Sun, S., and M.F. Wheeler, Projections of velocity data for the compatibility with transport, *Comput. Methods Appl. Mech. Engrg., 195*, 653-673, 2006.
[27] Thiele, M.R., Streamline simulation, In *P*roceedings of the 8th International Forum on Reservoir Simulation, Stresa / Lago Maggiore, Italy, 2005.

Part III

Supporting material

Paper G

Simulation of flow and transport in a discrete fracture-matrix system I. geostatistical generation of fractures on an aquifer analogue scale

*

* Submitted to *Water Resources Research*, 2008

Simulation of flow and transport processes in a discrete fracture-matrix system I. geostatistical generation of fractures on an aquifer analogue scale

A. Assteerawatt H. Hægland R. Helmig A. Bárdossy
H. K. Dahle

Abstract

Aquifer-analogue studies established in the petroleum industry have been widely used for characterizing fractured aquifer systems. Detailed analysis can be performed practically on an analogue scale and characteristics of fractured systems obtained on this scale can be upscaled to field scales. A discrete fracture-matrix model is an attractive alternative for studying on the analogue scale compared with single- and multi-continuum models since the effect of individual fractures can be explicitly investigated. The critical step for the discrete fracture model is the generation of a "representative" fracture network. In general, fracture networks are generated by describing fracture geometries in terms of statistical distribution and often neglecting the spatial variability. This tool is a so-called statistical fracture generator. In this study, we develop a geostatistical fracture generator which integrates statistical geometries and spatial characteristics in terms of a standardized variogram, neighborhoods and a fracture-cell density. Later the flow and transport behavior of a fracture-matrix system is investigated. We show that fracture networks generated by the GFG, to some extent, represent not only the included spatial characteristics but also a desired fracture-distance distribution (which is not considered in the GFG itself), and can better capture the flow and transport behavior of the fracture-matrix system (discharge, peak arrival time, and mean arrival time) than the fracture networks generated by the SFG. Hence, integrating the spatial characteristics and the statistical geometries in the GFG have improved the discrete fracture generation and the flow and transport behavior of the fractured system can be better predicted.

1 Introduction

For many countries worldwide, fractured rock systems have provided important natural resources such as petroleum, gas, water and geothermal energy. Many recent studies investigate the suitability of fractured systems as storage/disposal sites for high-level nuclear waste ([10, 11, 30, 12]). The resource exploitation and potential utilization have led to extensive studies with the aim of understanding, characterizing and finally predicting the behavior of fractured rock systems. Aquifer-analogue studies have been widely used for characterizing fractured systems ([2]). In the analogue studies, the detailed analysis of fractured systems such as borehole samplings, hydraulic measurements or exposed wall investigations can be performed practically, and the flow and transport properties obtained on the analogue scale can be upscaled to the field scale ([18]). On the large reservoir scale, the fractured system is considered as a single- or double-continuum, however, on the analogue scale, this assumption is not always valid. Long et al. [25] showed that the validity of considering a fractured system

as a continuum depended on the geometries of the fractured system, e.g. density and orientation. For the aquifer analogue studies, a discrete fracture model is an attractive alternative compared with a continuum model because there is no a priori assumption that the fractured system behaves as a continuum, and the effect of individual fractures can be explicitly investigated. The increased speed of computers nowadays makes even computationally demanding simulations, which is the main drawback of the discrete approach, feasible on the analogue scale. The critical step for the discrete model approach is the generation of a "representative" fracture network ([28]). In general, fracture network generators can be categorized as: a) deterministic fracture generators (DFG), and b) statistical fracture generators (SFG). The geometries of a fracture network (e.g. size, aperture, orientation and location) are determined exactly for the individual fractures in DFG, whereas they are represented by statistical distribution functions in SFG ([23, 19, 36, 13]). Due to the intensive data requirements in DFG, SFG is generally more practical. In the case where dominating fractures are known, a combination of SFG with DFG is required. However, SFG often fails to capture spatial variability and connectivity of fracture networks, which is related to interconnected flow paths that may dominate flow and transport processes in fractured systems ([8]). Spatial variability of fracture networks has been observed in many studies. La Pointe and Hudson [22] showed that fracture density and fracture orientation can follow a systematic spatial pattern. For a large scale problem, the attempt to incorporate spatial variability was presented by Long and Billaux [24]. They evaluated the experimental variogram of fracture traces from a drift wall and generated a two-dimensional heterogeneous fracture network which contained statistically homogeneous sub-domains. This concept was extended to generate a three-dimensional fracture network of circular-disc fractures by Billaux et al. [9]. However, on the analogue scale, assuming fracture geometries to be variable in the whole domain and homogeneous in sub-domains is not applicable because the domain size relative to the fracture size is relatively small.

The objective of this study is to develop a geostatistical fracture generator (GFG) which directly handles the statistical geometries as well as the spatial variability. We characterize spatial variability and connectivity of a fracture network from exposed walls (outcrops, tunnels or drifts) and consider these parameters in the geostatistical fracture generator (GFG). To our knowledge this is the first time that spatial connectivity of fractures is directly included in fracture generators. In the GFG, first a fracture network is generated by the statistical fracture generator (SFG) developed by Silberhorn-Hemminger [32] and then its spatial characteristics are included using a global optimization method known as Simulated Annealing (SA). A successful application of SA in optimizing spatial problems concerning fractured systems have been reported by several authors ([17, 4, 16, 34]). The flow and transport behavior of a fractured system are investigated to compare the fracture network generated by the GFG and the SFG. In this study, the fractured system describes a fracture network embedded in a surrounding matrix which is permeable to flow, although less permeable than the fractures by several orders of magnitude. This type of a fractured system is commonly called a fractured porous media or a fracture-matrix system (FMS).

The paper is organized as follows. In Section 2, we introduce the methodology to quantify spatial characteristics from exposed walls. In Section 3, the GFG based on the statistical geometries and the analyzed spatial variability is presented. A comparative study of fractured systems where fracture networks are generated by the GFG and by the SFG is performed in Section 4. In Section 5, we summerize the results and discuss future research work.

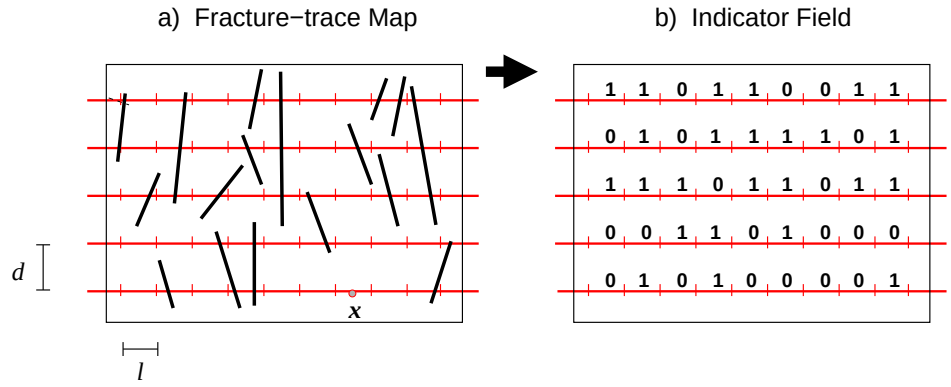


Figure 1: Procedure of evaluating an indicator field from a fracture-trace map.

2 Analysis of Spatial Characteristics

We quantify the spatial characteristics of the fracture network from fracture-trace maps of exposed walls by the *modified scanline technique* (MS) ([18]). In Figure 1, the scanline method is illustrated. First, a set of parallel and equally-distributed scanlines are overlayed on a fracture-trace map, where the direction of scanlines is arranged perpendicular to the main orientation of the fracture traces. The scanlines are divided into segments of equal length l. The fracture trace map is hence divided into a grid of cells, see Figure 1b. Then, along each scanline, the existence of intersection points between fracture traces and the scanline segment itself are recorded and described with an indicator variable $I(\boldsymbol{x})$:

$$I(\boldsymbol{x}) = \begin{cases} 1 & \text{intersection fracture-scanline exists (called fracture cell)} \\ 0 & \text{no intersection fracture-scanline exists (called matrix cell)} \end{cases} . \tag{1}$$

Here $\boldsymbol{x}$ is a discrete variable that represents the midpoint of a cell. In this way, the fracture trace map is transformed into an indicator field. The distance d between scanlines, and the length l of the scanline segment, are defined depending on the size of the fracture-trace maps and the characteristics of the fracture traces. Different scanline distances and scanline segments should be investigated in order to find suitable values ([18]).
The study site in this work is a field block of $8 \times 10 \times 2$ m located at Pliezhausen, Germany (see Figure 2). The field block is characterized by dense fractures and high porosity and permeability of the matrix. The fracture trace maps are recorded from a surface of five exposed walls by performing stereophotogrametric shooting ([18]). Figure 3 shows the fracture traces recorded on the five exposed walls: the north, the east, the south-east, the south-west and the west walls. Three main fracture clusters, one with almost horizontal orientation and two with almost vertical orientations, are identified from the 3D field block using stereographic projection, however only two main directions of fracture traces are observed on the 2D exposed walls: one horizontal, and one vertical. Since the scanline direction has to be perpendicular to the main direction of the fracture trace, two scanline directions are required. Two indicator fields are evaluated from each exposed wall: one for the horizontal scanline direction considering vertical fracture traces and another one for the vertical scanline direction considering horizontal fracture traces. According to the analysis of the Pliezhausen exposed walls presented in Silberhorn-Hemminger [32], the experimental variogram evaluated from the scanline distance $d = 0.10$ m (for the horizontal scanline) and $d = 0.20$ m (for the vertical

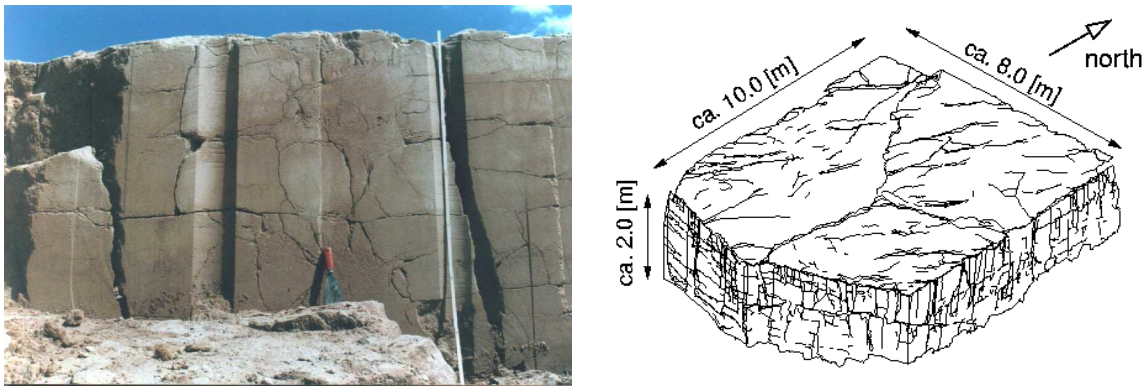


Figure 2: The field block and scanned view of the field block from the south-east (modified from Dietrich et al. [18]).

scanline) showed similar characteristic for either segment length of 0.04 or 0.10 m. Therefore, the segment length $l = 0.10$ m is selected in this study because it economizes computation time for our further spatial analysis.
The spatial variability is analyzed from indicator fields considering the vertical and the horizontal fracture trace. The first parameter is a standardized experimental variogram, which measures the average of an increment of values between two points, and at the same time, considers the variability of the indicator fields in terms of the variance σ^2:

$$\gamma_s(\boldsymbol{h}) \;=\; \frac{1}{2\sigma^2}\left(\frac{1}{n_h}\sum_{\alpha=1}^{n_h}\left[I(\boldsymbol{x}_\alpha+\boldsymbol{h}) \;-\; I(\boldsymbol{x}_\alpha)\right]^2\right)\,, \tag{2}$$

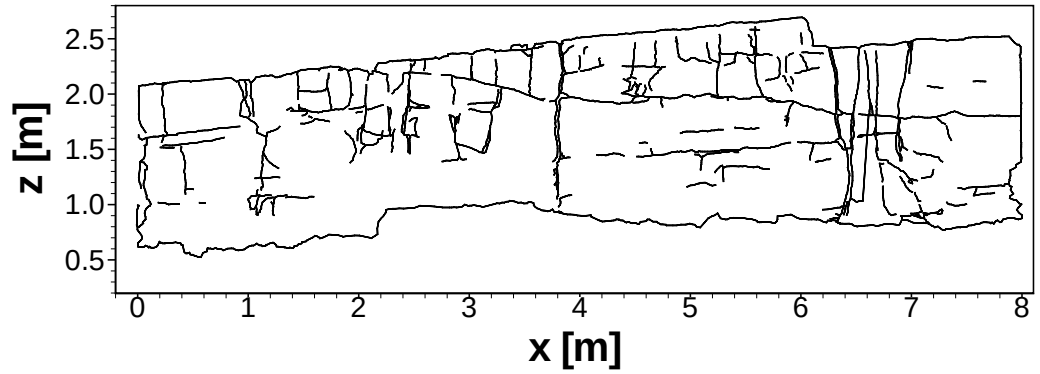
where the separation vector $|\boldsymbol{h}|$ is measured parallel to the main direction of the fracture trace (perpendicular to the scanline direction) and n_h is the total number of pairs of variables at a distance $|\boldsymbol{h}|$ apart. Since the indicator variables can have two possible outcomes 0 and 1, its variance is then described according to a Bernoulli trial, which relates to a trial that can have two possible outcomes. The variance of the indicator field is then given by

$$\sigma^2 \;=\; p(1-p)\,, \qquad \text{where} \qquad p \;=\; \frac{1}{n}\sum_{\alpha=1}^{n} I(\boldsymbol{x}_\alpha)\,, \tag{3}$$

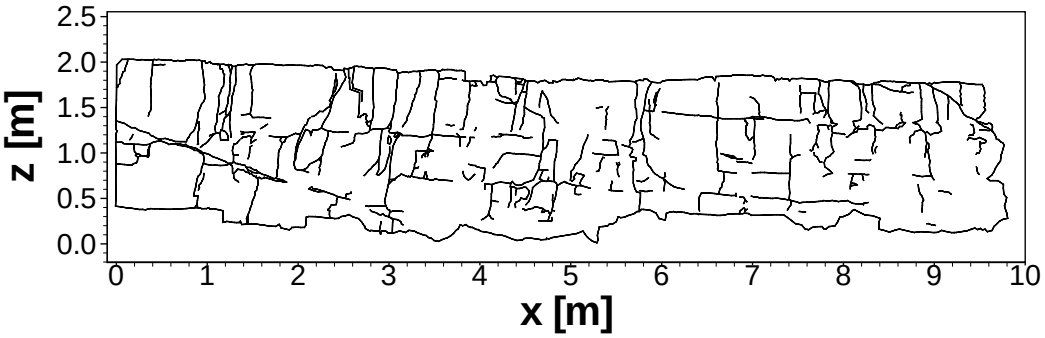
where p is the probability of being a fracture cell and n is the total number of cells.
Figure 4 shows that the standardized variograms of the five exposed walls increase within a specific distance $\boldsymbol{h}$, known as the correlation length or the range, and later reach a constant value, known as the sill. This means that a spatial dependency of the fracture trace exists within that range. Only the separation vector $\boldsymbol{h}$ parallel to the main direction of the fracture trace is considered in this paper; for $\boldsymbol{h}$ perpendicular to the fracture-trace direction, the experimental variograms of all exposed walls show only the nugget effect which means no spatial dependency, see Assteerawatt [3]. The average of the standardized experimental variograms of the five exposed walls are fitted with variogram models by VARIOWIN ([29]). The average experimental variograms show best fit with a combination of the exponential and the nugget variogram model ([14]):

$$\gamma(\boldsymbol{h}) \;=\; C_0 + C_1\,(1 \;-\; e^{\left(-\frac{|\boldsymbol{h}|}{a}\right)}) \qquad \text{for} \quad a \;>\; 0\,. \tag{4}$$

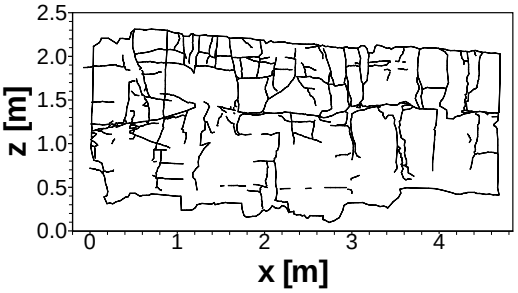
a) North exposed wall



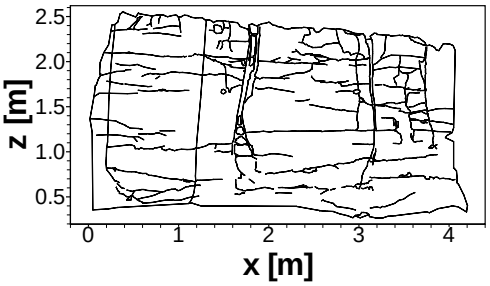
b) East exposed wall



c) South-east exposed wall



d) South-west exposed wall



e) West exposed wall

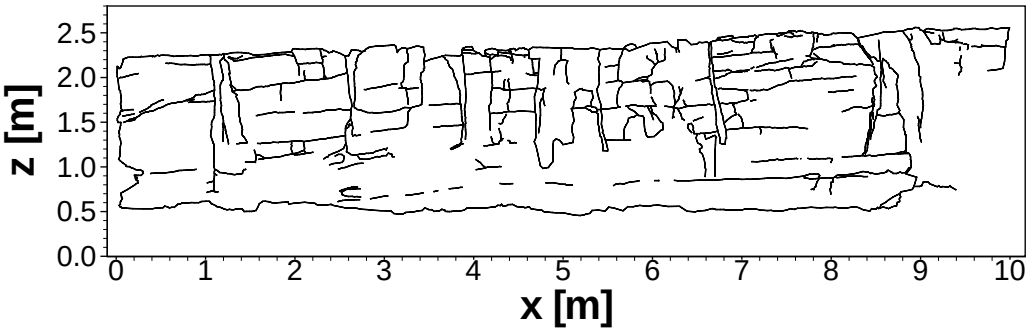


Figure 3: Fracture trace maps of the five exposed walls obtained from a Pliezhausen field block (modified from Dietrich et al. [18]).

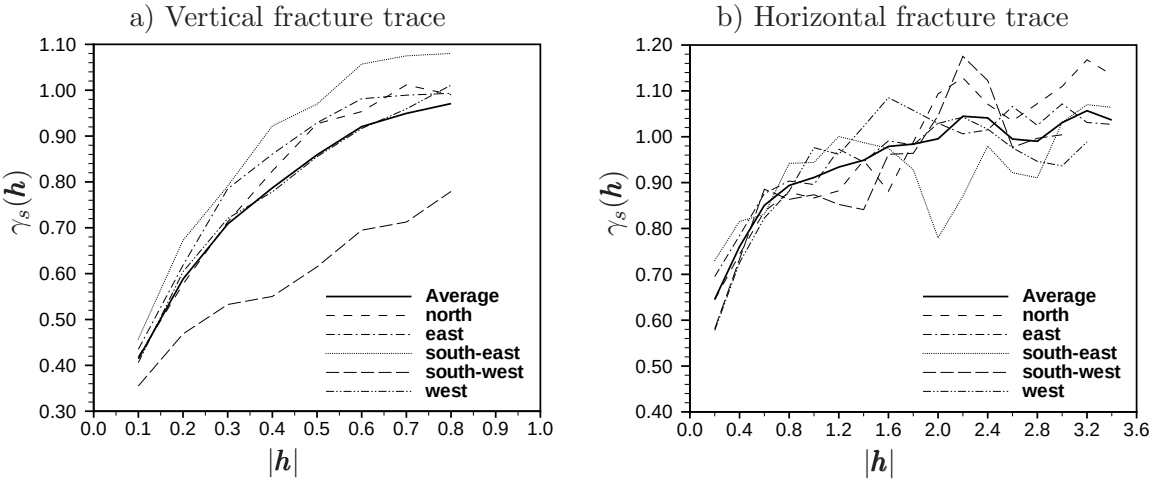


Figure 4: Standardized experimental variograms of the five exposed walls.

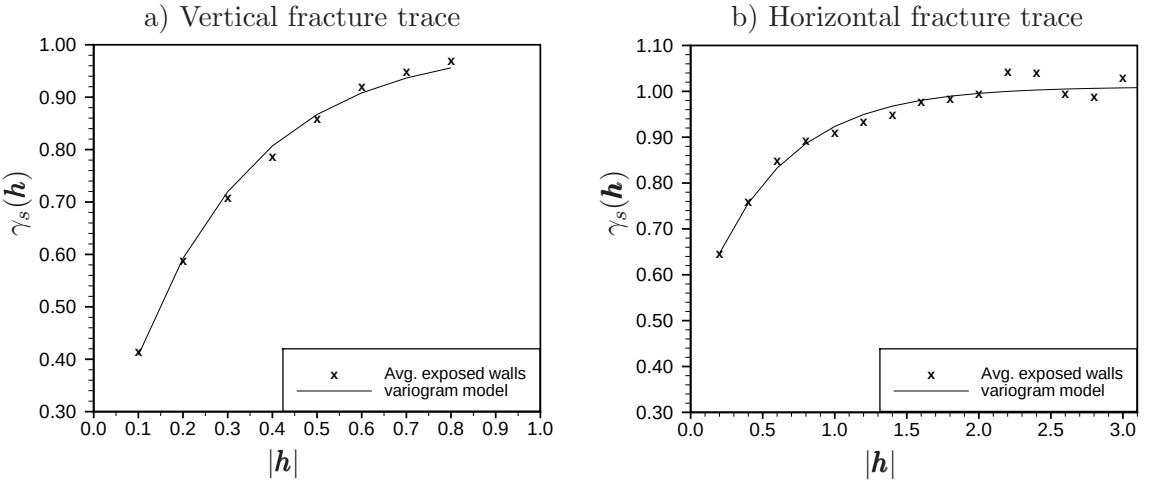


Figure 5: Variogram models from the average of the standardized experimental variograms.

The parameters C_0, C_1 and a, are 0.14, 0.86 and 0.80, respectively, for the horizontal-scanline indicator fields and 0.48, 0.52 and 1.67, respectively, for the vertical-scanline indicator fields. In this case, the sum of C_0 and C_1 is 1 because the experimental variograms are normalized by the variances. The best-fit variograms for the average values of all five standardized experimental variograms are shown in Figure 5. The standardized variogram only consider the spatial variability in the direction of the fracture trace, therefore, the additional parameters called neighborhoods are used to characterize the spatial connectivity of the fracture traces related to other directions. The fracture neighborhood N_f and the matrix neighborhood N_m describes the probability of finding a fracture cell or a matrix cell in the eight adjacent cells of a centered cell $I(\boldsymbol{x}_\alpha)$:

$$\begin{aligned} N_f(k) &= \tfrac{1}{n_f} \textstyle\sum_\alpha^n I(\boldsymbol{x}_\alpha^k) I(\boldsymbol{x}_\alpha) \ , \\ N_m(k) &= \tfrac{1}{n_f} \textstyle\sum_\alpha^n I(\boldsymbol{x}_\alpha^k)(1 - I(\boldsymbol{x}_\alpha)) \ . \end{aligned} \tag{5}$$

Here n is the total number of cells, n_f is the total number of fracture cells, $\boldsymbol{x}_\alpha^k$ is the adjacent cell located in direction k of $\boldsymbol{x}_\alpha$. The directions k, illustrated in Figure 6, are represented by the digits 0 to 7.

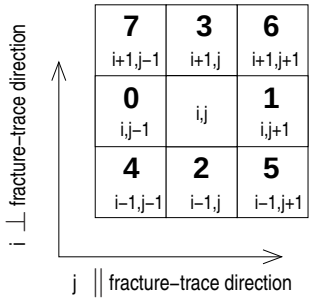


Figure 6: Illustration of neighborhood directions in connection with the digits 0 to 7.

The neighborhoods N_f and N_m of the five exposed walls from the vertical and the horizontal fracture trace are presented in Figure 7 and 8, respectively. Due to the values assigned in Equation (1), a high value of neighborhoods corresponds to a large number of fracture cells. A high value of N_f is clearly observed in the fracture-trace direction ($k = 2, 3$) as expected because neighboring cells of a fracture cell in the fracture direction have a tendency to be fracture cells. The N_m are significantly lower in the direction of fracture traces meaning that neighboring cells of a matrix cell in the fracture direction tend to be a matrix cell. The values of the neighborhoods in the other directions are related to the number of the fracture and the matrix cells found in those directions. If a center cell is a fracture cell and a neighboring cell in a non-fracture-trace direction ($k \neq 2, 3$) is also a fracture cell, this could indicate existence of a fracture trace in that direction, hence the spatial connectivity of fracture traces is related to the values of the neighborhoods.

Due to the weathering process acting on the exposed outcrop, the largest number of vertical fractures is observed on the top. For representing this fracture structure, we set up a parameter called fracture-cell density $H(z)$, where the z-coordinate increases towards the top of the outcrop. It is defined by the number of fracture cells relative to the total number of cells along each horizontal scanline. The fracture-cell density is only considered for the vertical fracture traces in the horizontal scanline direction. In Figure 9, the results from the five

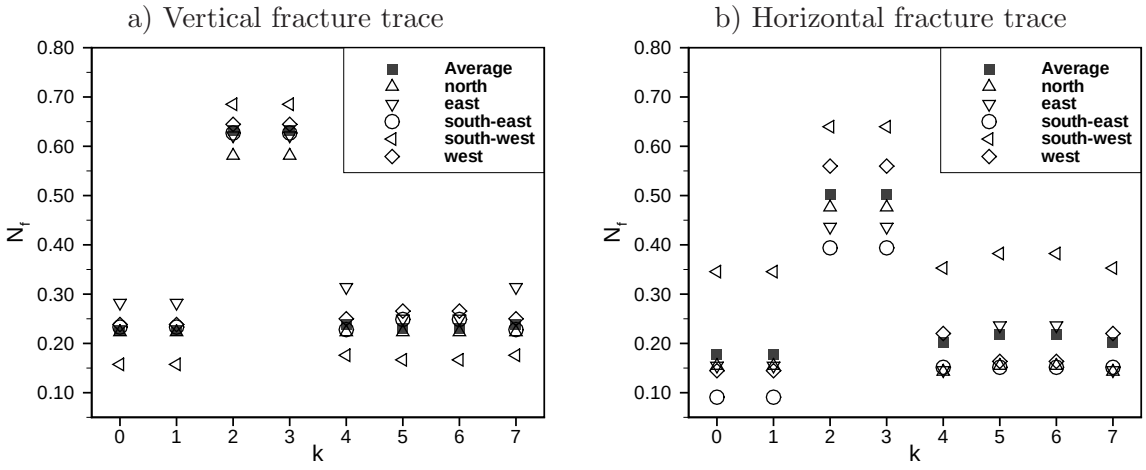


Figure 7: Fracture neighborhood of the five exposed walls.

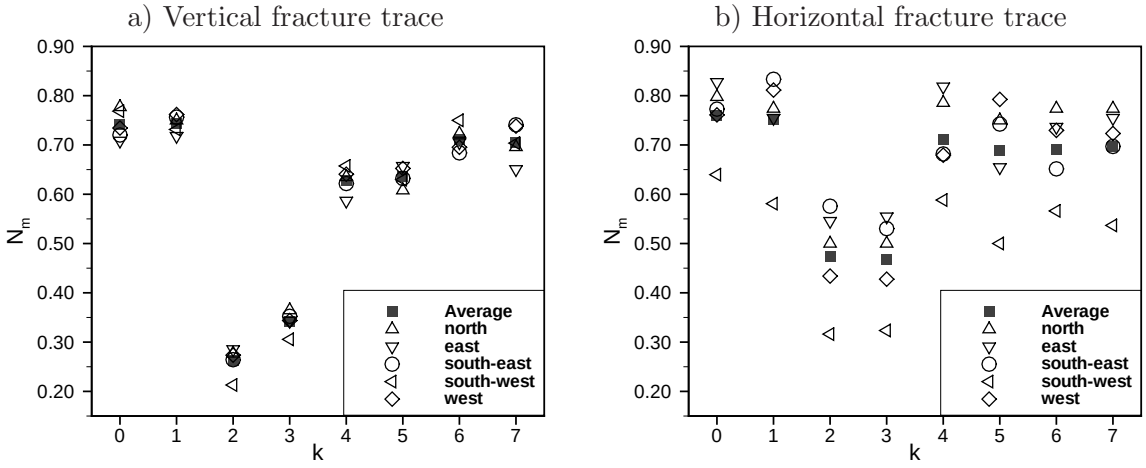


Figure 8: Matrix neighborhood of the five exposed walls.

exposed walls clearly show an increase of the number of fracture cells with increased vertical position z, which is in agreement with our observation.

The probability p expressed in Equation (3) is included as one of the parameters in the GFG in order to control the number of fracture cells and matrix cells. It is assumed to be constant and the average p evaluated from the horizontal- and vertical-scanline indicator fields are 0.2112 and 0.2377, respectively.

The spatial parameters which are analyzed from the outcrop block are considered as spatial characteristics of the outcrop fracture network. The geostatistical fracture generation (GFG) is carried out in the next section by integrating the spatial characteristics which are discussed in this section and the statistical geometries of the fracture network such as the fracture orientation, the fracture density and the fracture size.

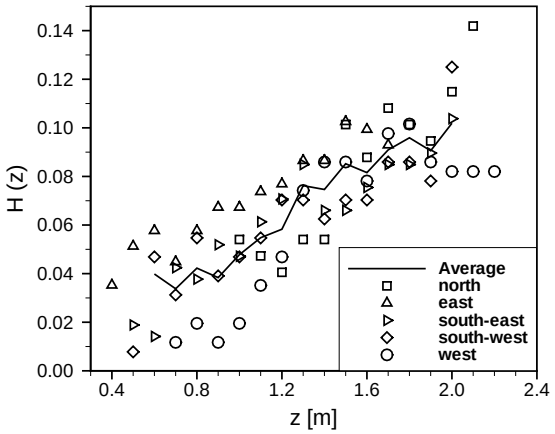


Figure 9: Fracture-cell density of the five walls for the vertical fracture traces.

3 Geostatistical Fracture Generation

The success of a simulated annealing (SA) as a global optimization method in integrating multiple characteristics of discrete fractures ([4, 35]), and its flexibility to incorporate additional information (if needed), make the SA attractive as a methodology for fracture-network generation. Additionally, the SA can locate good approximations to global optimum of a given objective function in a large search space ([1]). The application of SA in optimization problems concerning fractured systems have been reported by several authors ([26, 16, 34]). The objective function of the SA is defined as the difference between a set of reference properties from a desired configuration and from a candidate realization. The SA starts at an initial state, and the system is modified randomly to a new state. The new state is accepted if the change decreases the objective function or if the change increases the objective function, it is accepted with the probability

$$p_{acc} = \begin{cases} 1 & \text{if } O(S_2) \leq O(S_1) \\ e^{\left(-\frac{O(S_2)-O(S_1)}{T}\right)}, & \text{if } O(S_2) > O(S_1)\ . \end{cases} \tag{6}$$

Here the constant parameter T, called annealing temperature, is used for controlling the acceptance of the new state, and $O(S_1)$ and $O(S_2)$ are the values of the objective function of the current state (S_1) and the new state (S_2), respectively. At each temperature, the perturbation is repeated for a large number of iterations M before decreasing the temperature. The temperature T is then reduced linearly to a new temperature T_{new} with the cooling factor α, $T_{new} = \alpha\, T$. When the temperature T becomes lower, the probability of accepting changes which cause high objective functions, becomes smaller. This allows the problem to converge. The SA is used as a methodology in the GFG for integrating the desired spatial characteristics in the generated domain. The GFG was implemented in the existing 3D-fracture generator *FRAC3D* developed by Silberhorn-Hemminger [32]. The FRAC3D originally offers two different approaches: the deterministic fracture generator (DFG) and the statistical fracture generator (SFG). The DFG requires exact information of a fracture network and the SFG assumes that a fracture network can be described by theoretical distribution of its geometries such as orientation, size and density. The SFG starts with generating a random midpoint of a fracture and determines the location of the fracture from its orientation and size. The

newly generated fracture is included into the global list of fractures until the desired fracture density is reached. This approach is based on univariate statistics and does not include any information about the spatial variability. Optimizing the fracture distance, which is defined by distances between two directly adjacent fractures measured along a reference line, is an optional step in the SFG that can be used to include spatial information in the generation of a fracture network ([32]).

The GFG takes into account the statistical geometries and the spatial variability of the indicator fields (the standardized experimental variogram, the neighborhoods, the fracture-cell density and the variance), see Assteerawatt [3]. The GFG starts by generating an initial state of the fracture network using the SFG. Thus, the generated fracture network obeys the statistical geometries (size, orientation and density). By defining several investigated cross-sections in the generated domain, the spatial parameters can be evaluated from the investigated cross-sections. The objective function $O(k)$ of the randomly generated system is defined as the sum of normalized differences between the spatial parameters of the target fracture network ($\hat{\gamma_s}(\boldsymbol{h})$, $\hat{N}_f(j)$, $\hat{N}_m(j)$, $\hat{H}(z)$ and $\hat{\sigma}^2$) and of the generated fracture network at state k ($\gamma_s^k(\boldsymbol{h})$, $N_f^k(j)$, $N_m^k(j)$, $H^k(z)$ and ${\sigma^k}^2$) from all N_e investigated cross sections:

$$\begin{aligned} O(k) \;=\; & \sum_{i=1}^{N_e}\Big(w_1 \sum_{j=1}^{n_{\gamma_s}} \frac{|\hat{\gamma_s}(\boldsymbol{h}_j) - \gamma_s^k(\boldsymbol{h}_j)|}{\hat{\gamma_s}(\boldsymbol{h}_j)} \;+\; w_2 \sum_{j=0}^{7} \frac{|\hat{N}_f(j) - N_f^k(j)|}{\hat{N}_f(j)} \\ & + w_3 \sum_{j=0}^{7} \frac{|\hat{N}_m(j) - N_m^k(j)|}{\hat{N}_m(j)} \;+\; w_4 \sum_{j=1}^{n_H} \frac{|\hat{H}(z_j) - H^k(z_j)|)}{\hat{H}(z_j)} \\ & + w_5 \frac{|\hat{\sigma}^2 - {\sigma^k}^2|}{\hat{\sigma}^2}\Big)\,, \end{aligned} \tag{7}$$

where w_i is the weighting function and $\sum w_i = 1$. By introducing the weighting function, the influence of each spatial parameter on the objective function can be controlled. After evaluating the objective function of the current state from Equation (7), the fracture network is modified to a new configuration and a new objective function value is calculated. Comparing the two objective functions, the new network is accepted with the acceptance criteria mentioned in Equation (6). The fracture network is modified at each iteration step by adding/removing one fracture, or randomly selecting one fracture and changing its geometries (location, orientation or shape). All fractures are convex polygons consisting of four to seven edges. The fracture shape is modified by adding/removing a corner point under the constraint of a total number of the corner points, or by moving a corner point. The perturbation is carried out for M iterations and then the temperature T is reduced related to the cooling factor α. Finally, the SA stops when the objective function remains unimproved after a couple of temperature steps, or the minimum temperature T_{stop} is reached. The configuration with a minimum objective function is regarded as the solution to the problem of finding a realization which has spatial characteristics closest to the target fracture network. The major difficulty in applying SA is that there is no obvious analogy for defining the initial temperature T_0, the number of iterations M, and the cooling factor α at each temperature step. In accordance with Bárdossy [5], the value of the initial temperature T_0 is selected so that the initial acceptance probability of a new stage is approximately 0.80 for assuring that many "bad" configurations which do not improve the objective function are accepted in the beginning. The number of iterations M is defined as half of the total number of fractures, to ensure a high probability of the generated fractures to be involved in the exchange processes.

Typical values of the cooling factor α for moderately slow cooling rates are 0.90 through 0.99. In this work, the cooling factor of 0.90 is selected and remains constant for all temperature steps.

4 Comparison of Geostatistical and Statistical Fracture Generation

In this section, the fracture networks generated by the GFG and the SFG are compared. The distribution functions of fracture geometries observed on the Pliezhausen block are summarized in Table 1.

	Parameters of Distribution Function		
Orientation:			
Fisher Distribution	$f(\theta,\phi) = \frac{\kappa}{4\pi \sinh\kappa}\, e^{[\kappa\,(\sin\theta\,\sin\alpha\,\cos(\phi-\beta)+\cos\theta\,\cos\alpha)]\,\sin\theta}$		
	Azimuth (A)	Dip (D)	Spherical Aperture (ω)
	$A = 360° - \phi$	$D = \theta - 90°$	$\omega = \arcsin\sqrt{2\frac{1-1/n}{\kappa}}$
Set1	201°	85°	11.22°
Set2	146°	7°	12.05°
Set3	229°	8°	10.20°
Fracture Trace:			
Erlang-2 Distribution	$f(x) = \lambda^2 x e^{-\lambda x}$		
	lambda (λ)		
horizontal	-5.37		
vertical	-3.93		
Fracture Distance:			
Exponential Distribution	$f(x) = \lambda e^{-\lambda x}$		
	lambda (λ)		
	-4.57		
Fracture Density:			
Uniform Distribution	$f(x) = 11.34 \quad m^2/m^3$		

Table 1: Statistics of the fracture geometries observed from the Pliezhausen field block.

Two study cases of SFG (SFG-A and SFG-B) and two study cases of GFG (GFG-A and GFG-B) are considered. The two cases of SFG are set up according to two different assumptions on defining the fracture size. As the fracture size cannot be measured directly from borehole samples or exposed wall surveys, it has to be approximated from the fracture trace observed on the exposed walls by assuming a relation between the fracture size and the observed trace length. In both SFG-A and SFG-B rectangular fractures are assumed. Hence the fracture size can be described by distributions for two of its side-lengths. In SFG-A, the fractures are assumed to have the size directly defined by the fracture-trace distribution shown in Table 1. The length and the width of the horizontal fractures are described by the horizontal trace distribution, whereas the height and the width of the vertical fractures are given by the vertical and the horizontal trace distribution. In SFG-B, the fracture-size distribution is adjusted until the same fracture-trace distribution as for the field block is obtained. The parameter lambda in the erlang-2 distribution describing the horizontal and vertical length

distribution in the SFG-B are 4.28 and 1.70 respectively, instead of 5.37 and 3.93 as in the SFG-A. Reducing the parameter lambda leads to an increase of the fracture size. After the fracture network of the specified fracture geometries (size, orientation and density) has been obtained, the fracture distance is optimized according to the distribution observed from the field block. In the two cases of GFG (GFG-A and GFG-B), an initial configuration of the fracture network is taken from the SFG, where the fracture-distance optimization option is not used, (GFG-A from SFG-A and GFG-B from SFG-B) and then the fracture network is modified by the SA until its spatial parameters similar to those observed from the field outcrop.

Fracture networks are compared in this section by considering two different aspects; one is the structure of the generated fracture networks, and another is the flow and transport behavior of the fracture-matrix systems (FMS). From a stochastic point of view, it is possible to generate multiple realizations of fracture networks such that each realization represents the specified descriptions, however, no single realization can exactly match the real system. The structure of the fracture network and the behavior of the FMS can be quantitatively predicted from the ensemble average which is obtained only when the number of realizations is large enough to assure convergent results. We intend to study but not to predict the characteristics and the behavior of the fractured system, therefore, twenty realizations (five from each study case) are generated. Even though the number of realization seems to be very low, the results of the four cases, which are presented later, show significant differences. For comparing different cases, parameters of each study case are evaluated by averaging over the five exposed walls of each realization and again over all five realizations. The south-west cross-section of one realization from each study case is shown in Figure 10.

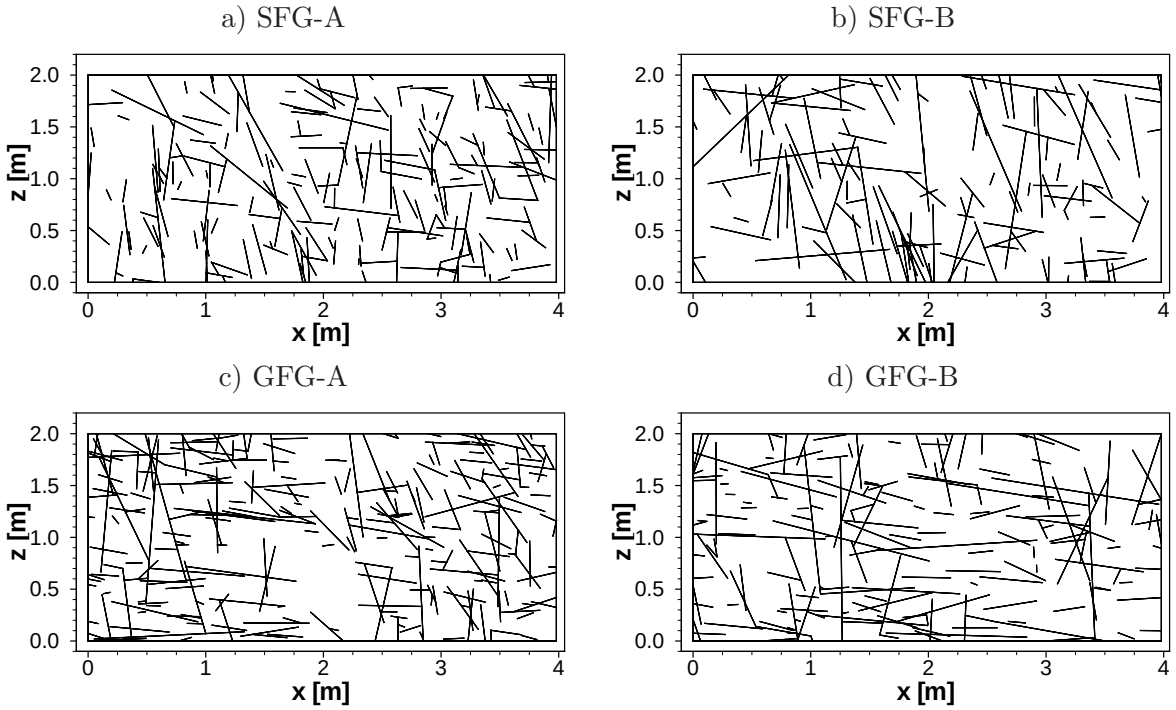


Figure 10: Fracture-trace maps of the south-west exposed wall obtained from the four study cases.

4.1 Structure of a Fracture Network

The fracture networks generated from the four study cases (the SFG-A, the SFG-B, the GFG-A and the GFG-B) are compared with the field block by focusing on the fracture size (fracture trace-length) and the spatial characteristics. The only spatial structure of the fracture network which is considered in the SFG is the fracture distance. On the other hands, the GFG takes into account the spatial characteristics such as standardized experimental variogram, the neighborhoods and the fracture-cell density.

Fracture trace

The average of the cumulated fracture-trace distribution is presented in Figure 11. The result from the SFG-A shows that approximating the fracture size directly from the fracture trace results in a too short trace length distribution, which means the fracture size is underestimated in this case. The SFG-B enlarges the fracture size to fit the fracture-trace distribution, therefore its results show a good agreement with the outcrop. The trace length distribution of the GFG-A changes slightly compared with the SFG-A after optimizing the spatial characteristics. For the GFG-B, the trace-length distribution is smaller than for the SFG-B, which means that optimizing spatial characteristics in this case leads to a reduction of the fracture size.

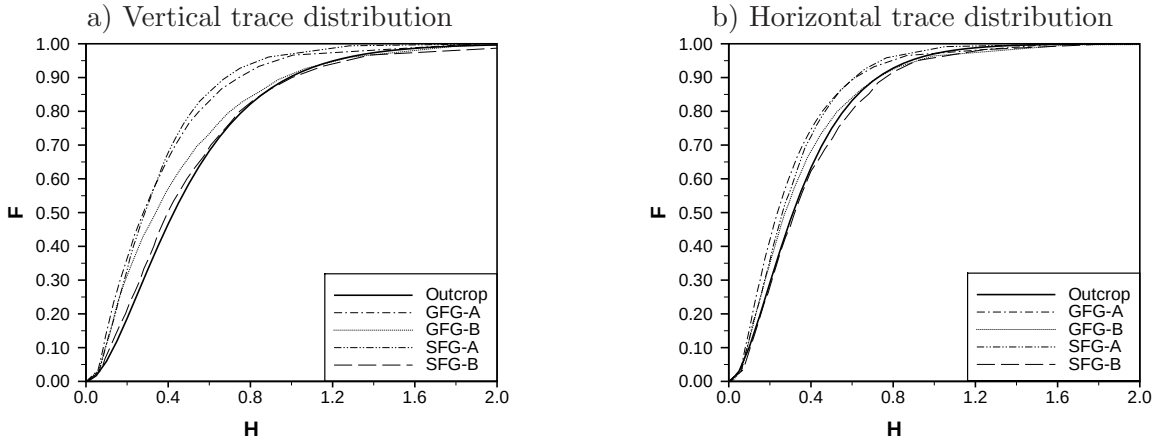


Figure 11: Comparison of the cumulative distribution functions of trace length averaged from the five exposed walls of fracture networks generated from the four study cases.

Fracture distance

Figure 12 shows the cumulated distribution of the fracture distance of the four study cases compared with the field block. The SFG-A and the SFG-B optimize the fracture distance, therefore, they show better agreement with the field value than the GFG-A and the GFG-B. A larger size of fractures in the SFG-B and the GFG-B compared with the SFG-A and the GFG-A means that a less number of fractures are given for the same fracture density, therefore, the distance between two adjacent fractures tends to become longer in the SFG-B and the GFG-B than in the SFG-A and the GFG-A. We do not include the fracture distance as one of the spatial parameter in the geostatistical fracture generator, however, this parameter could be included (if needed).

Standardized experimental variogram

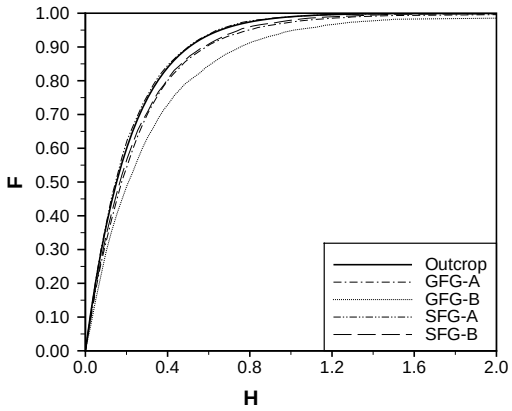


Figure 12: Comparison of the cumulative distribution functions of fracture distance averaged from the five exposed walls of fracture networks generated from the four study cases.

The standardized experimental variogram of the GFG-A and the GFG-B are closer to the outcrop target value than the SFG-A and the SFG-B for the vertical and horizontal fracture-trace directions (Figure 13).

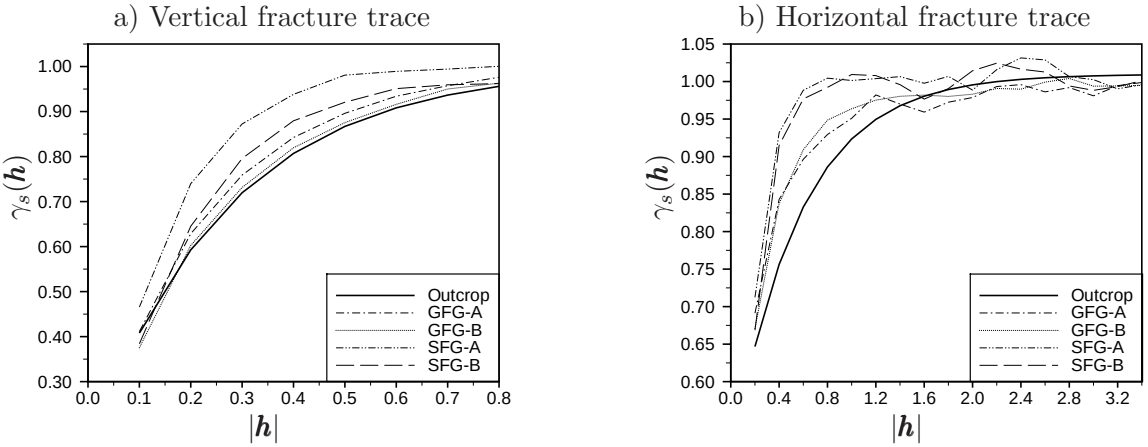


Figure 13: Comparison of the standardized experimental variograms averaged from the five exposed walls of fracture networks generated from the four study cases.

Neighborhoods

The neighborhoods of a fracture and a matrix cell for the vertical fracture trace of the GFG-A and the GFG-B are closer to the field block than the SFG-A and the SFG-B (Figure 14a and 15a). However, the neighborhoods for the horizontal fracture trace of the GFG-A and the GFG-B closely match only in the fracture direction ($k = 2, 3$) but not in the other directions, where the SFG-A and the SFG-B show better results (Figure 14b and 15b). The reason could be that the neighborhoods for the horizontal fracture trace are varied in a range broader than for the vertical fracture trace and tend to be directionally dependent on the south-west walls (see Figure 7b and 8b). Correct representation of a parameter which has this peculiarity might need a more complicated approach. The directional dependence of the neighborhoods is observed

from all realizations generated from the GFG-A and the GFG-B. As shown in Figure 16 and 17, the N_f and the N_m from the field block are close to the values from the north, the east and the west exposed walls but not to the values from the south-east and south-west exposed walls.

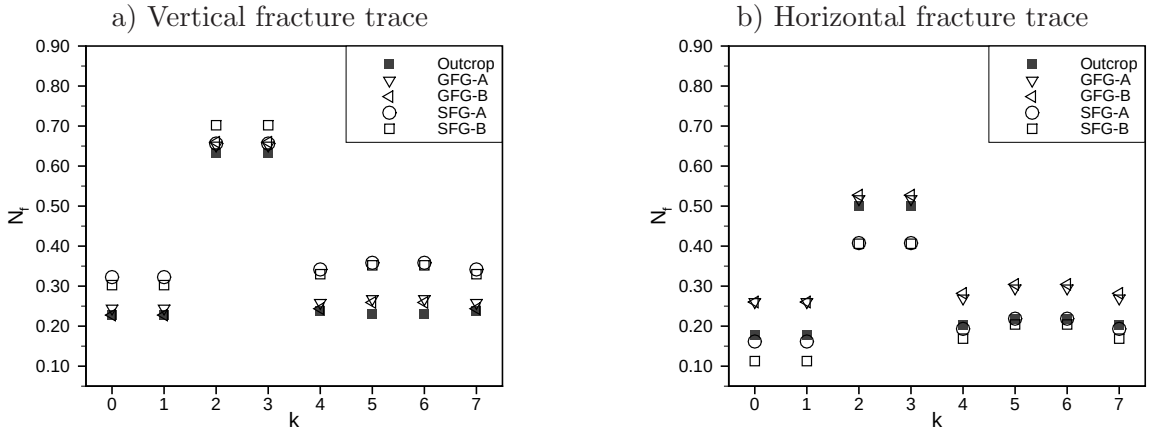


Figure 14: Comparison of the fracture neighborhood averaged from the five exposed walls of fracture networks generated from the four study cases.

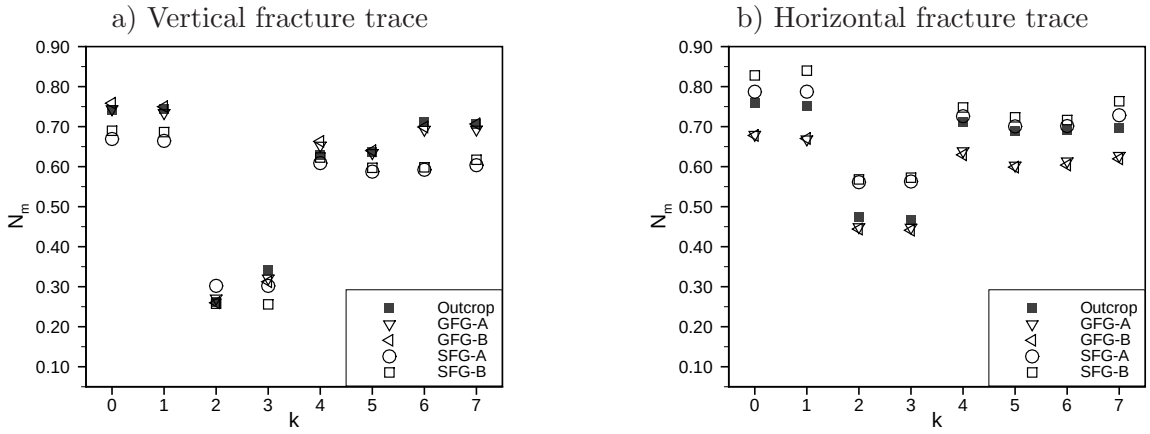


Figure 15: Comparison of the matrix neighborhood averaged from the five exposed walls of fracture networks generated from the four study cases.

Fracture-cell density

The fracture-cell density H evaluated from all cases is shown in Figure 18. The greater number of vertical fractures on the top according to the field outcrop, is only observed from the GFG-A and the GFG-B. The SFG-A and the SFG-B show relatively constant H due to equally distributed fracture cells.

Even though the geostatistical approach (the GFG-A and the GFG-B) does not consider the cumulative distribution of the fracture distance, it still reflects the cumulative distance fairly well. On the other hand, the statistical approach can adequately reproduce the neighborhoods, but not the standardized variogram and the fracture-cell density.

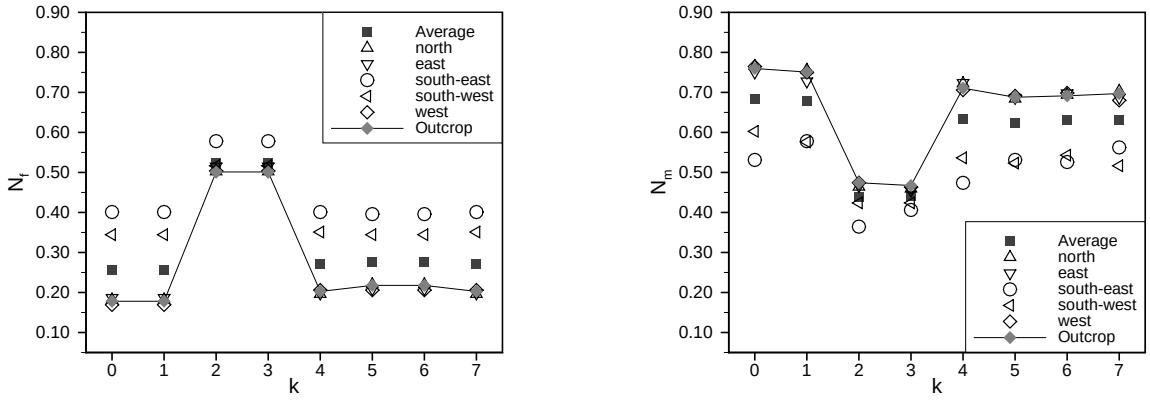


Figure 16: Neighborhoods of one realization of the GFG-A from vertical-scanline indicator fields of the five exposed walls, scanline segment length $l = 0.10\,\mathrm{m}$ and scanline distance $d = 0.20\,\mathrm{m}$.

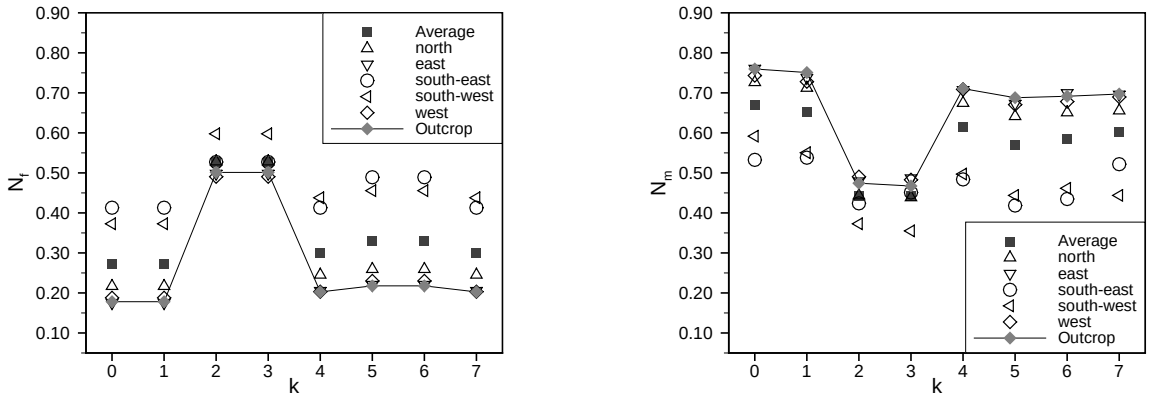


Figure 17: Neighborhoods of one realization of the GFG-B from vertical-scanline indicator fields of the five exposed walls, scanline segment length $l = 0.10\,\mathrm{m}$ and scanline distance $d = 0.20\,\mathrm{m}$.

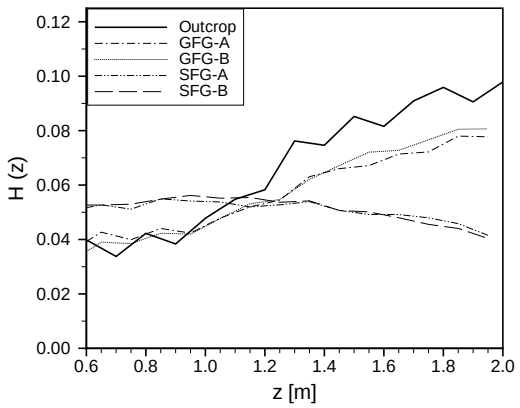


Figure 18: Comparison of the fracture-cell density averaged from the five exposed walls of fracture networks generated from the four study cases for the vertical fracture trace.

4.2 Flow and Transport in a Fracture-matrix System

A comparative study of flow and transport behavior is carried out using the numerical model MUFTE-UG (MUltiphase Flow, Transport and Energy Model - Unstructured Grids) ([15] and [6]).

4.2.1 Formulation of Flow and Transport Equation

For simplicity, steady-state flow of an incompressible single phase in a nondeformable matrix is considered, and source/sink terms are neglected. The continuity equation based on mass conservation can be described as:

$$\nabla \cdot \boldsymbol{q} = 0 \; . \tag{8}$$

Here the velocity $\boldsymbol{q}$ is defined by Darcy's law as a function of the pressure p and the permeability tensor $\boldsymbol{K}$:

$$\boldsymbol{q} = -\frac{\boldsymbol{K}}{\mu}(\nabla p - \rho \boldsymbol{g}) \; , \tag{9}$$

where μ is the fluid viscosity, ρ is the fluid density, and $\boldsymbol{g}$ is the gravitational vector with absolute value equal to the gravitational constant g. Neglecting the gravitational effect in Equation (9) results in

$$\boldsymbol{q} = -\frac{\boldsymbol{K}}{\mu}\nabla p. \tag{10}$$

For fractures a scalar permeability is assumed given by the parallel-plate concept ([33]):

$$K \; = \; \frac{b^2}{12} \; , \tag{11}$$

where b is the fracture aperture.
The governing equation for solute conservative transport process without source/sink terms is given as:

$$\frac{\partial c}{\partial t} + \nabla \cdot (\boldsymbol{v}_s c - \boldsymbol{D}\nabla c) = 0 \; . \tag{12}$$

Here the seepage velocity $\boldsymbol{v}_s$ is a function of the effective porosity ϕ and the Darcy velocity:

$$\boldsymbol{v}_s \; = \; \frac{1}{\phi}\,\boldsymbol{q} \; . \tag{13}$$

The hydrodynamic dispersion $\boldsymbol{D}$ in a two-dimensional case where the transport direction follows the coordinate axis is given as:

$$\boldsymbol{D}_{ij} \; = \; \frac{v_i v_j}{\|\boldsymbol{v}\|}(\alpha_l - \alpha_t) + \delta_{ij}(\alpha_t \|\boldsymbol{v}\| + D_m); \, , \tag{14}$$

where D_m is the molecular diffusion, α_l and α_t are the longitudinal and transversal dispersion lengths, v_x and v_y are the components of seepage velocity in the longitudinal and transversal direction, and the Kronecker delta δ_{ij} is unity for i = j and zero otherwise.

4.2.2 Simulation of Fracture-Network System

The two-dimensional simulations are performed on the south-west cross-sections for the twenty realizations, five from each of the four study cases. These results are compared with a simulation of the scanned south-west wall from the outcrop (Figure 3d). The domain is discretized with a triangular mesh, with lower-dimensional fractures. Then a vertex-centered finite- volume method with upwinding formulation is used to solve Equation 12, see Reichenberger et al. [31] and for details. Boundary conditions are described in Figure 19 and model parameters are shown in Table 2. In this study, a highly advective transport is considered, therefore very low longitudinal and transversal dispersivity are assumed. This advective-dominated transport allows the direct comparison with the results of a streamline method in the accompanying paper ([20]).

		matrix	**fracture**
Permeability, K	$[m^2]$	$1.0 \cdot 10^{-13}$	$8.33 \cdot 10^{-10}$
Eff. porosity, ϕ	[-]	0.13	0.30
Long. dispersivity, α_l	$[m]$	$1.0 \cdot 10^{-9}$	0.0
Trans. dispersivity, α_t	$[m]$	$1.0 \cdot 10^{-9}$	0.0
Diffusion coeff., D_m	$[m^2 s^{-1}]$	$1.0 \cdot 10^{-9}$	$1.0 \cdot 10^{-9}$
Aperture, d	$[m]$	-	$1.0 \cdot 10^{-4}$

Table 2: Flow and transport parameters for all simulations

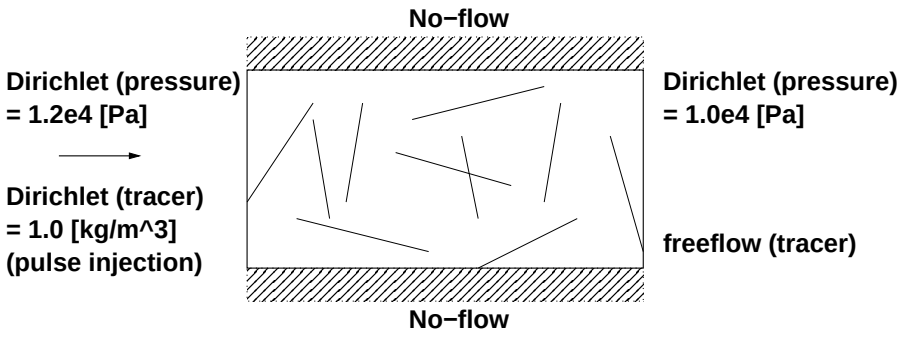


Figure 19: Boundary conditions for flow and transport simulations

The flow and transport behavior of the different case studies are compared with respect to the total outflow Q and the characteristics of the breakthrough curve (BTC) such as a peak mass flux $\dot{m}_p$, a peak arrival time t_p and a mean arrival time $\bar{t}$. The peak mass flux and its arrival time are directly observed from BTCs. The travel time $\bar{t}$ is evaluated from the moment μ_i as:

$$\bar{t} = \frac{\mu_1}{\mu_0} \qquad \text{where} \quad \mu_i = \int_0^\infty t^i c(t) dt \tag{15}$$

Here $c(t)$ is the total mass concentration [kg/s] leaving the domain at time t. The BTCs obtained from transport simulations are shown in Figure 20 and the total discharge Q and the BTC characteristics are summarized in Table 3. The deviation of the simulation results

from the scanned wall is shown in Figure 21, where the standardized value (SV) is defined as

$$SV = \frac{P_r - P_{sw}}{P_{sw}} . \tag{16}$$

Here, P_r is the average of each investigated parameter over the realizations and P_{sw} is the parameter from the scanned south-west wall.

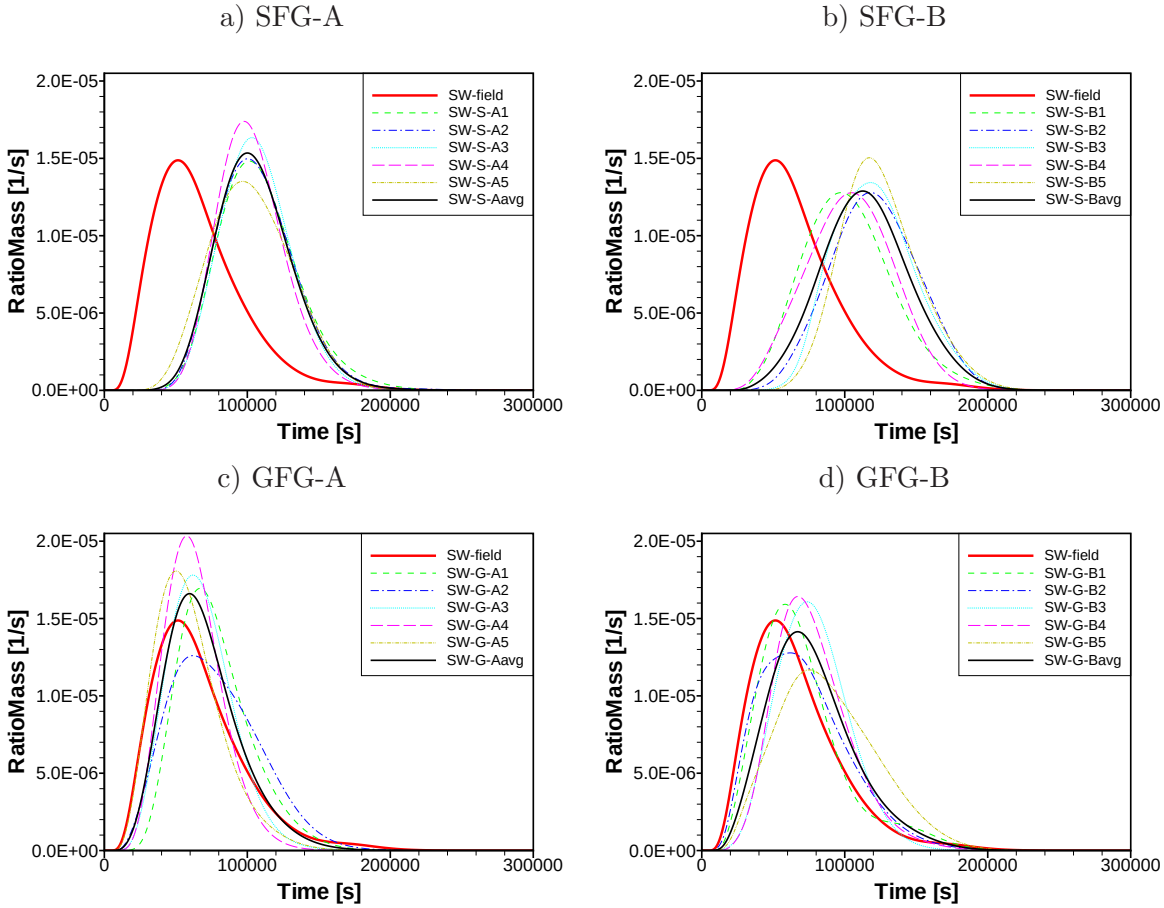


Figure 20: Comparison of breakthrough curves of the south-west walls from different realizations of the four study cases.

The flow and transport behavior of the field outcrop can be, to some extent, represented by GFG-A and GFG-B, however, SFG-A and SFG-B show clearly slower processes (see Figure 20 and Table 3). Compared to the simulated field block, Q, $\bar{t}$ and t_p from GFG-A and GFG-B are closer to the field block than those from SFG-A and SFG-B (see Figure 21). By considering the spatial structures of the fractured system, the connectivity of fractures can be represented. Hence the discharge, the peak arrival time and the mean arrival time can be better approximated. However, connected and preferential flow paths lead to significant variation of the peak mass flux and of the shape of the BTCs. Therefore, the average behavior of the fracture-matrix systems requires a simulation performed on large number of realizations.

The influence of the fracture size observed from the case with and without fracture-trace fitting remains uncertain. The GFG-B shows closer agreement to the outcrop compared with

	Q $[kg\ m^{-3}]$	$\dot{m}_p$ $[kg\ s^{-1}]$	t_p $[s]$	$\bar{t}$ $[s]$
Outcrop	$1.72 \cdot 10^{-5}$	$1.49 \cdot 10^{-5}$	$5.12 \cdot 10^{4}$	$6.64 \cdot 10^{4}$
SFG-A	$1.03 \cdot 10^{-5}$	$1.53 \cdot 10^{-5}$	$1.00 \cdot 10^{5}$	$1.06 \cdot 10^{5}$
SFG-B	$9.36 \cdot 10^{-6}$	$1.29 \cdot 10^{-5}$	$1.13 \cdot 10^{5}$	$1.15 \cdot 10^{5}$
GFG-A	$1.73 \cdot 10^{-5}$	$1.66 \cdot 10^{-5}$	$5.97 \cdot 10^{4}$	$6.90 \cdot 10^{4}$
GFG-B	$1.56 \cdot 10^{-5}$	$1.41 \cdot 10^{-5}$	$6.72 \cdot 10^{4}$	$7.74 \cdot 10^{4}$

Table 3: Results of flow and transport simulations of the south-west wall obtained form the field outcrop and average values from four study cases.

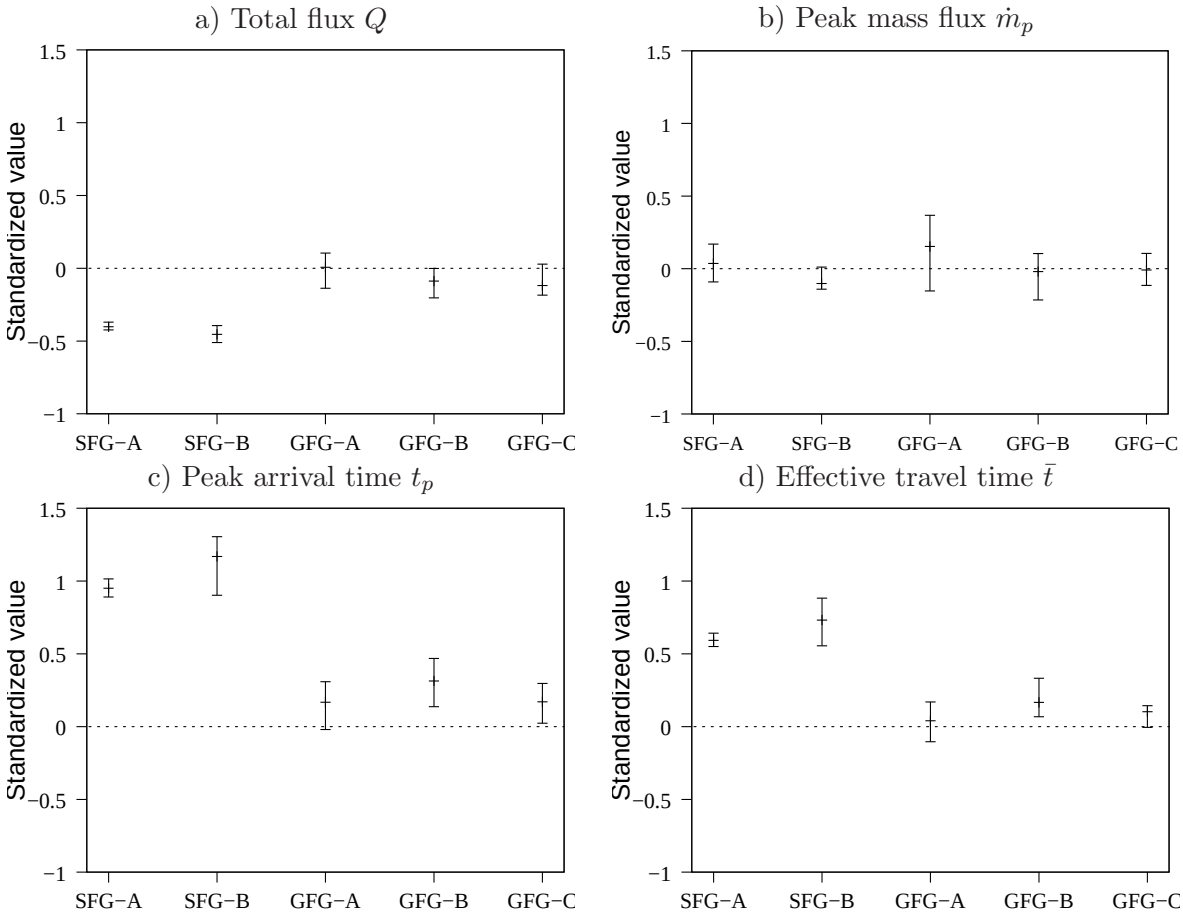


Figure 21: Average and extreme values (min./max.) of flow and transport simulations of the south-west wall of the study cases compared with the results obtained from the field outcrop simulation (dashed line).

the GFG-A for $\dot{m}_p$ but not for Q, t_p and $\bar{t}$. Comparing with the results from the outcrop, the SFG-B seems to be slightly better than the SFG-A for all cases. Investigation on more realizations could be necessary to draw a conclusion about the effect of the fracture size and the average behavior of the systems.
In addition, the influence of numerical diffusion can be noticed by comparing the BTCs of

the scanned south-west wall for different mesh sizes (measured in terms of the average length of the sides of a grid cell): 0.01, 0.02, 0.05, 0.10 and 0.20 m, corresponding to 93547, 23389, 4277, 1177 and 528 grid vertices, respectively. The numerical diffusion in a FMS leads to a surprising result of the BTCs. When the grid sizes become smaller, the variances of the curves become larger and the peak value decreases (see Figure 22). This is due to a physical diffusion caused by the strong heterogeneity between the fractures and the matrix. As shown in Figure 23a and 23b for a fine mesh of 0.02 m, when the injected solute reaches the fractures, it is transported quickly along the fractures and the solute remaining in the matrix requires longer time to transport out of the FMS. On the contrary, for solute transport on a coarse mesh of 0.20 m shown in Figure 23c and 23d, the effect of fast-flowing in the fractures and slow-flowing in the matrix is smeared out over the cross-section perpendicular to the flow direction due to the numerical diffusion. The FMS tends to behave more homogeneous. To guarantee the accuracy of the scheme, a small grid size is necessary, hence, the computational time increases.

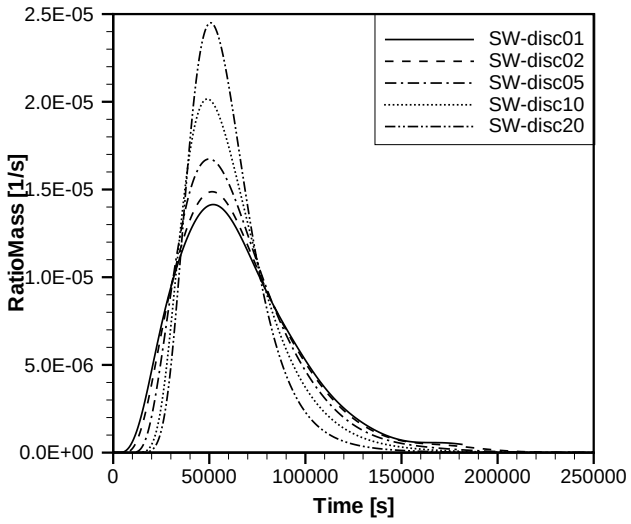
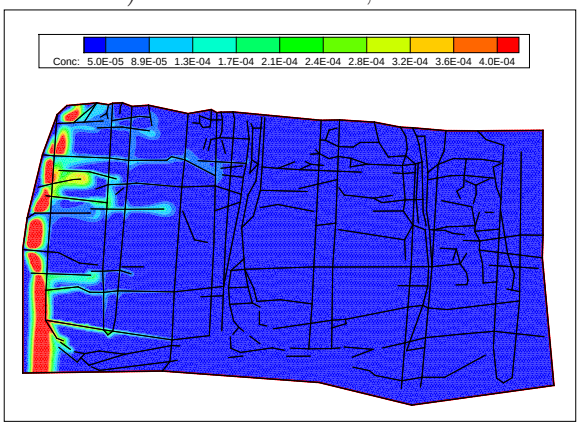


Figure 22: Comparison of breakthrough curves for different mesh sizes: 0.01, 0.02, 0.05, 0.10 and 0.20 m for the south-west exposed wall.

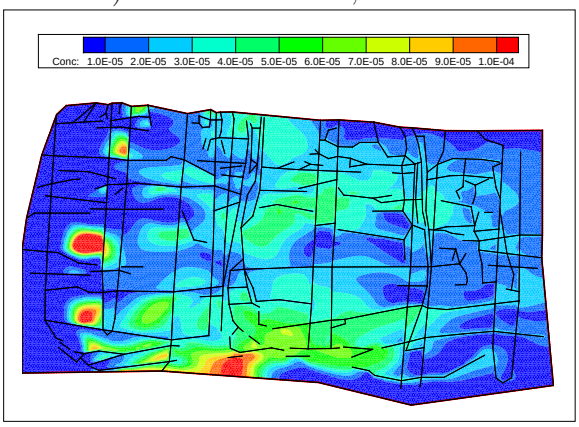
5 Conclusion and Outlook

A geostatistical fracture generator (GFG) which integrates statistical geometries and spatial characteristics has been presented in this work. By applying a modified scanline technique, fracture-trace maps of exposed walls are transformed to indicator fields on which the spatial characteristics are quantified in terms of the standardized variogram, the neighborhoods and the fracture-cell density. Simulated annealing is selected as our methodology for integrating the spatial characteristics in the GFG. We have shown here that fracture networks generated by the GFG not only can reproduce the chosen spatial characteristics, but also, to some extent, can represent the cumulative distribution of the fracture distance observed from the field which is not included in the GFG. The fracture networks created by the statistical fracture

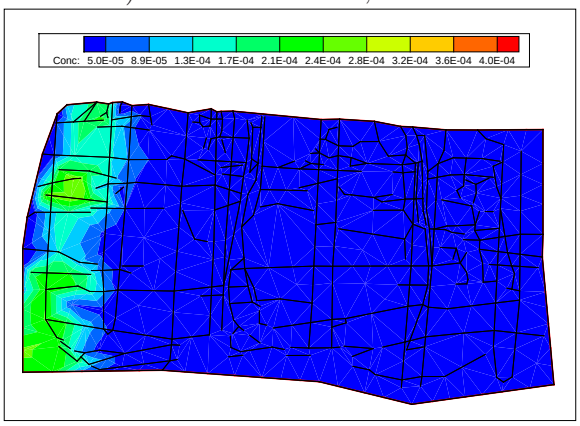
a) Mesh size 0.02 m, t 200 s



b) Mesh size 0.02 m, t 32000 s



c) Mesh size 0.20 m, t 200 s



d) Mesh size 0.20 m, t 32000 s

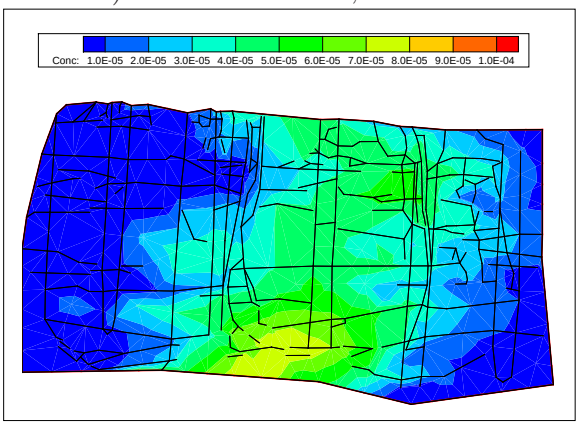


Figure 23: Concentration distribution for the south-west wall at time 200 and 32000 s using mesh sizes 0.02 (fine grid) and 0.20 (coarse grid) m.

generator (SFG) show good results for the fracture-distance distribution and reasonably good results for the neighborhoods, however not for the standardized variogram and the fracture-cell density. Further, strong influence of the spatial characteristics was clearly demonstrated in flow and transport simulations. The GFG can better capture the system behavior such as the discharge, the peak arrival time and the mean arrival time than the two study cases from the SFG. Integrating the spatial characteristics and the statistical geometries in the GFG have improved the discrete fracture generation and, therefore, the behavior of the fractured system can be better predicted. Considering only the fracture distance as a spatial structure and the statistical geometries as in the SFG is not sufficient in this case to generate a "representative" fracture network. Additional spatial parameters such as the standardized variogram, the neighborhoods and the fracture-cell density must also be considered.
The flow and transport process in a three-dimensional fracture-matrix system should be further investigated, since the connectivity of fractures in 3D becomes even more complex than in 2D. Considering this effect might lead to an effort on extending the geostatistical fracture generator to take into account additional spatial characteristics.
Due to the strong heterogeneity between fractures and the surrounding matrix and the numerical diffusion, accurate results of flow and transport simulations in a fracture-matrix system can only be obtained on a fine mesh. This means that an approach that performs fast with less numerical diffusion is essential. In an accompanying paper ([20]), we present an application of streamline tracing on a fracture-matrix system. The advantages of the streamline method is that it is extremely fast compared with the standard finite volume scheme (which is used here) and, at the same time, does not suffer from numerical diffusion.

6 Acknowledgments

This work was supported by the International Research Training Group "Non-Linearities and Upscaling in Porous Media (NUPUS)" financed by the German Research Foundation (DFG) and in parts by the Norwegian Research Council, StatoilHydro, and Norske Shell under grant no. 178013/13. The authors would like to thank Wolfgang Nowak for his valuable comments on the manuscript.

References

[1] Aarts, E., and J. Korst, *Simulated annealing and Boltzmann machines*, John Wiley & Sons, 1989.

[2] Alexander, J., *A discussion on the use of analogues for reservoir geology*, In ASHTON, M. (Ed.), *Advances in Reservoir Geology*, Geological Society, Special Publications, n. 69, 175-194, 1992.

[3] Assteerawatt, A., *Flow and Transport Modelling of Fractured Aquifers based on a Geostatistical Approach*, (http://elib.uni-stuttgart.de/opus/volltexte/2008/3639/), PhD Thesis, Universität Stuttgart, Institute of Hydraulic Engineering, 2008.

[4] Soares, A., A. Brusco, and C. Guimaraes, Simulation of Naturally Fractured Fields, *Geostatistics Wollongong '96 (Quantitative Geology and Geostatistics)*, Kluwer Academic Publishers, 1997.

[5] Bárdossy, A., Generating precipitation time series using Simulated Annealing. *Water Resour. Res., 34(7)*, 1737 - 1744, 1998.

[6] Bastian, P., K. Birken, S. Lang, K. Eckstein, N. Neuss, H. Rentz-Reichert, and C. Wieners, UG - A Flexible Software Toolbox for solving partial differential equations, *Computing and Visualization in Science*, 1997.

[7] Bear, J., *Dynamics of Fluids in Porous Media*, Academic Press, 1972.

[8] Berkowitz, B., Characterizing flow and transport in fractured geological media: A review, *Adv. Water Resour., 25*, 861-884, 2002.

[9] Billaux, D., J. Chiles, K. Hestira, and J. Longa, Three-dimensional statistical modelling of a fractured rock mass–an example from the Fanay-Augères mine, *Int. J. Rock Mech. Min. Sci. Geomech. Abstr. 26(3,4)*, 281-299, 1989.

[10] Bodvarsson, G., and Y. Tsang, Special issue: Yucca mountain project, *J. Contam. Hydrol. , 38*, 1-425, 1999.

[11] Bodvarsson, G., C. Ho, and B. Robinson, Special issue: Yucca mountain project, *J. Contam. Hydrol. , 62-63*, 1-750, 2003.

[12] Cacas, M., E. Ledoux, G. de Marsily, A. Barbreau, P. Calmels, B. Gaillard, and R. Margritta, Modeling Fracture Flow With a Stochastic Discrete Fracture Network: Calibration and Validation 2. The Transport Model, *Water Resour. Res., 26(3)*, 491-500, 1990.

[13] Cacas, M., E. Ledoux, G. de Marsily, B. Tillie, A. Barbreau, E. Durand, B. Feuga, and P. Peaudecerf, Modeling Fracture Flow With a Stochastic Discrete Fracture Network. Calibration and Validation 1. The Flow Model, *Water Resour. Res., 26(3)*, 479-489, 1990.

[14] Chilés, J., and P. Delfiner, Geostatistics: Modeling Spatial Uncertainty, *Wiley series in probability and mathematical statistics*, John Wiley & Sons, Inc., 1999.

[15] Helmig, R., H. Class, R. Huber, H. Sheta, R. Ewing, R. Hinkelmann, H. Jakobs, and P. Bastian, Architecture of the Modular Program System MUFTE-UG for Simulating Multiphase Flow and Transport Processes in Heterogeneous Porous Media, *Mathematische Geologie, 2*, 123-131, 1998.

[16] Day-Lewis, F., P. Hsieh, and S. Gorelick, Identifying fracture-zone geometry using simulated annealing and hydraulic-connection data, *Water Resour. Res., 36(7)*, 1707-1721, 2000.

[17] Deutsch, C., and P. Cockerham, Practical considerations in the application of simulated annealing to stochastic simulation, *Math. Geol., 26(1)*, 67-82, 1994.

[18] Dietrich, P., R. Helmig, M. Sauter, H. Hötzl, J.Köngeter, and G. Teutsch, *Flow and Transport in Fractured Porous Media*, Springer-Verlag, 2005.

[19] Dverstorp, B., and J. Andersson, Application of the Discrete Fracture Network Concept with Field Data: Possibilities of Model Calibration and Validation, *Water Resour. Res., 25(3)*, 540-550, 1989.

[20] Hægland, H., A. Assteerawatt, R. Helmig, and H.K. Dahle, Simulation of flow and transport in discrete fracture-matrix system: II. Efficient and accurate streamline approach, *Water Resour. Res.*, submitted, 2008.

[21] Kirkpatrick, S., C. D. Gelatt Jr., and M. Vecchi, Optimization by Simulated Annealing, *Science, 220*, 671-680, 1983.

[22] La Pointe, P., and J. Hudson, Characterization and interpretation of rock mass joint patterns, *Special paper 199, Geological Society of America*, 1985.

[23] Long, J., *Investigation of equivalent porous medium permeability in networks of discontinuous fractures*, PhD Thesis, Berkley, 1983.

[24] Long, J., and D. Billaux, From Field Data to Fracture Network Modeling: An Example Incorporating Spatial Structure, *Water Resour. Res., 23(7)*, 1201-1216, 1987.

[25] Long, J., J. Remer, C. Wilson, and P. Witherspoon, Porous Media Equivalents for Networks of Discontinuous Fractures, *Water Resour. Res., 18(3)*, 645-658, 1982.

[26] Mauldon, A., K. Karasaki, S. Martel, J. Long, M. Landsfeld, A. Mensch, and S. Vomvoris, An inverse technique for developing models for fluid flow in fracture systems using simulated annealing, *Water Resour. Res., 29(11)* 3775-3789, 1993.

[27] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, Equations of state calculations by fast computing machines, *J. Chem. Phys., 21*, 1087-1092, 1953.

[28] National Research Council, *Rock Fractures and Fluid Flow*, National Academic Press, 1996.

[29] Pannatier, Y., *VARIOWIN: Software for spatial data analysis in 2D*, Springer-Verlag, 1996.

[30] Poteri, A., D. Billaux, B. Dershowitz, J.J. Gomez Hernandez, V. Cvetkovic, A. Hautojärvi, D. Holton, A. Medina, and A. Winberg, Final report of the TRUE Block Scale project. 3. Modelling of flow and transport, *Technical report, Swedish Nuclear Fuel and Waste Management Company (SKB)*, TR-02-15, 2002.

[31] Reichenberger, V., H. Jakobs, P. Bastian, and R. Helmig, A mixed-dimensional finite volume method for two-phase flow in fractured porous media, *Adv. Water Resour., 29(7)*, 1020-1036, 2006.

[32] Silberhorn-Hemminger, A., *Modellierung von Kluftaquifersystemen: Geostatistische Analyse und deterministisch – stochastische Kluftgenerierung*, PhD Thesis, Mitteilungen H. 114 , Institut für Wasserbau, Universität Stuttgart, 2002.

[33] Snow, D.T., Anisotropic permeability of fractured media, *Water Resour. Res., 5(6)*, 1273-1289, 1969.

[34] Tran, N., Simulated annealing technique in discrete fracture network inversion: optimizing the optimization, *Comput. Geosci., 11(3)*, 249-260, 2007.

[35] Tran, N., Z. Chen, and S. Rahman. Integrated conditional global optimisation for discrete fracture network modelling, *Comput. Geosci., 32(1)*, 17-27, 2006.

[36] Wollrath J., Ein Strömungs- und Transportmodell für klüftiges Gestein und Untersuchungen zu homogenen Ersatzsystemen, *Technical Report 28*, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, 1990.

Paper H

A method to improve the mass balance in streamline methods

*

* SPE Reservoir Simulation Symposium, 2007. Paper SPE 106250

SPE 106250

A Method to Improve the Mass Balance in Streamline Methods

Vegard Kippe, SINTEF, **Håkon Hægland**, University of Bergen, and **Knut–Andreas Lie**, SINTEF

This paper was prepared for presentation at the 2007 SPE Reservoir Simulation Symposium held in Houston, Texas, U.S.A., 26–28 February 2007.

Abstract

During the last decades, streamline methods have emerged as highly efficient simulation tools that are well-suited for e.g., history matching and simulation of large and complex reservoir models. Streamline methods are based on a sequential solution procedure in which pressure and fluid velocities are computed by solving a pressure equation on a grid in physical space and the fluid transport is computed by solving 1-D transport problems along streamlines. The sequential Eulerian-Lagrangian procedure is the key to the high computational efficiency of streamline methods. On the other hand, it necessitates mapping of saturations (or fluid compositions) back and forth between the Eulerian pressure grid and the Lagrangian streamlines. Unfortunately, this introduces mass-balance errors that may accumulate in time and in turn yield significant errors in production curves.

Mass-balance errors might be reduced by considering higher-order mapping algorithms, or by increasing the number of streamlines. Since the computational speed scales linearly with the number of streamlines, it is clearly desirable to use as few streamlines as possible. Here we propose a modification of the standard mapping algorithm that: (i) improves the mass-conservation properties of the method and (ii) provides high-accuracy production curves using few streamlines.

Mass conservation is improved by changing quantities in the transport equation locally, and we show that these modifications do not significantly affect the global saturation errors as long as a sufficient number of streamlines is used. Moreover, we propose an adaptive strategy for ensuring adequate streamline coverage. The efficiency and accuracy of the modified streamline method is demonstrated for Model 2 from the Tenth SPE Comparative Solution Project. Highly accurate production curves (compared to reference solutions) are obtained in less than ten minutes using one processor on a standard (Intel Core 2 Duo) desktop computer.

Introduction

Streamline simulation has experienced increasing industry interest and rapid technology development in recent years and is now a very efficient alternative to traditional flow modelling by numerical methods such as finite differences or finite volumes. Modern streamline methods can be used to compute complex flow physics such as compressible three-phase models with full PVT, multicomponent models or dual-porosity models (Thiele et al., 1997; Crane et al., 2000; Di Donato and Blunt, 2004). Still, streamline simulation is most efficient for simplified physical models and engineering queries based on the 80-20 principle: 80% of the answer in 20% of the time available (Thiele, 2005). In particular, due to its low memory requirements and high computational efficiency, streamline simulation today offers the opportunity to solve outstanding engineering queries that might otherwise be difficult or impossible to address using other approaches.

Streamline simulators are particularly suitable for solving large and geologically complex models, where the fluid flow is dictated primarily by heterogeneities in rock properties (permeability, porosity and faults/fractures), well positions, and phase mobilities. The typical application is for production regimes involving fluid displacement, e.g., water flood or gas injection. Other mechanisms, like capillary effects and expansion-driven flows, may be modelled, but not with the same degree of accuracy and efficiency. Primary examples of application are flow simulations on multimillion geocellular models of complex heterogeneity, and repeated simulations on equiprobable geological realisations to quantify sensitivity of model parameters and uncertainties in prediction forecasts. Generally, streamline

simulators are progressively being used more by operating companies as an alternative to traditional reservoir simulators in several reservoir engineering workflows, including: screening of enhanced recovery projects, rapid sensitivity studies, history matching, uncertainty assessment, upscaling, flood optimization, or simulation studies of sector or full-field models.

The computational setup within a streamline simulator can be briefly described as follows. First, the pressure distribution over a conventional 3-D grid is computed in order to determine the trajectories of 1-D streamlines that represent flow-paths. Next, the material balance equations can be transformed in terms of the so-called time-of-flight along a streamline and split into two parts, namely the part along the streamline and the part in the direction of gravity. These 1-D equations are then solved by an appropriate numerical method and the resulting saturation or concentration values are mapped back onto the 3-D grid. In each time-step, the velocity field is recomputed, which implies that streamline trajectories will change in time for dynamic flow conditions. For a more in-depth description of streamline simulation and an overview of the literature in this field, we refer the reader to the upcoming textbook by Datta-Gupta and King (to appear) or to the survey papers by Thiele (2005) and King and Datta-Gupta (1998).

The underlying mathematical formulation is both the strength and the weakness of streamline simulation. The operator splitting and the Lagrangian spatial discretization, which are fundamental assumptions of streamline methods, are the keys to obtaining high efficiency:

- The operator splitting used to decouple the computation of the velocity field (i.e., pressure) and the fluid transport has the effect that the size of the pressure steps is dictated by the flow dynamics, and not by the spatial (finite-difference) discretization. For e.g., water flood problems, this usually means that velocity fields and streamlines only need to be updated infrequently.
- The 1-D transport problems along streamlines and gravity lines can be solved very efficiently such that the computational complexity of the transport step scales linearly with the number of streamlines and the number of cells traversed by each streamline.
- The number of streamlines typically required to obtain an acceptable accuracy increases linearly with the number of active cells.

These three points, together with the existence of near-linear complexity linear solvers for the pressure equation (Stüben, 2000), imply that streamline simulation scales (almost) linearly with model size, may be very memory efficient, and offers a natural potential for parallel implementation. However, it is also evident that streamline simulation will loose its high efficiency for flows with a very strong coupling between the pressure and the mass transport equation.

Similarly, it is clearly desirable to use as few streamlines as possible to ensure efficient flow simulation. On the other hand, the set of streamlines should be representative and sufficiently dense to ensure accurate prediction of flow patterns and production responses, and to limit errors in the mass balance. Lack of mass conservation is a problem of particular concern to reservoir engineers, and in this paper we will try to analyse the lack of mass conservation and suggest methodological improvements that will strongly improve the mass balance. This will in turn allow a significant reduction in the number of streamlines required to ensure highly accurate production curves.

The rest of the paper is organized as follows: In the next two sections we define our model problem and describe what we shall refer to as our "standard" or "original" streamline method. The mass-balance problems are then illustrated with an example, and we utilize a description of the streamline spatial discretization given by Jimenez et al. (2005) to explain the problem. We propose a change of the original streamline method, and demonstrate that the modified approach improves the mass balance and gives accurate production curves using very few streamlines for a large and complex reservoir model. We then study the performance for various flow conditions on a very simple model and propose a strategy for ensuring adequate streamline coverage, before demonstrating applicability to a history-matching problem with more than a million grid blocks and 69 producers. Some final remarks then concludes the paper.

Model Problem

Since our focus in this paper is on the mass-balance properties of streamline methods, we will consider a simplified model for water flooding. That is, we assume immiscible, incompressible two-phase flow and disregard gravity and capillary forces. Our flow model then consists of an elliptic pressure equation

$$\nabla \cdot \mathbf{u} = q_t, \qquad \mathbf{u} = -\lambda_t(S)\mathbf{K}\nabla p, \tag{1}$$

and quasilinear hyperbolic transport equation

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot (f_w(S)\mathbf{u}) = q_w. \tag{2}$$

The primary unknowns in the coupled system (1)–(2) are the pressure p, the total (Darcy) velocity $\mathbf{u}$, and the water saturation S. The underlying porous rock formation is modelled in terms of the absolute permeability $\mathbf{K}$ and the porosity ϕ, which henceforth are assumed to depend on the spatial variable only. Finally, $\lambda_t = \lambda_w + \lambda_o$ denotes the total mobility, where the mobility of each phase, λ_j is given as the relative permeability k_{rj} of phase j divided by the phase viscosity μ_j $(j = o, w)$, and $f_w = \lambda_w/\lambda_t$ is the fractional flow of water.

The Streamline Method

The streamline method is based on a sequential solution procedure. First the known initial saturation distribution is used to compute the mobilities $\lambda_t(S)$ in (1), after which the pressure equation can be solved to give total velocity $\mathbf{u}$ and pressure distribution p. Next, the total velocity $\mathbf{u}$ is kept fixed in (2), while the saturation is advanced a given time step. The new saturation values are used to update the mobilities in (1), the pressure equation is solved again, and so on.

Instead of discretizing and solving (2) directly on a grid, a streamline method decouples the three-dimensional equation into multiple one-dimensional equations along streamlines by introducing the time-of-flight variable,

$$\tau(s) = \int_0^s \frac{\phi(\zeta)}{|\mathbf{u}(\zeta)|}\mathrm{d}\zeta, \tag{3}$$

which is the time it takes a passive particle to travel a distance s along a streamline. In differential form (3) becomes,

$$\frac{\partial \tau}{\partial s} = \frac{\phi}{|\mathbf{u}|} \quad \Longleftrightarrow \quad \mathbf{u}\cdot\nabla\tau = \phi. \tag{4}$$

Moreover, we have that $\partial/\partial\tau \equiv \mathbf{u}\cdot\nabla$, which combined with (4) can be used to rewrite the saturation equation (2) as a one-dimensional equation to be solved along each streamline,

$$\frac{\partial S}{\partial t} + \frac{\partial f_w}{\partial \tau} = 0. \tag{5}$$

The solution to the full three-dimensional problem (2) is obtained by tracing numerous streamlines in the domain, mapping the initial saturation distribution from the 3-D pressure grid to the one-dimensional streamlines, and then solving (5) along each streamline. Afterwards, the new saturation values along streamlines must be mapped (or averaged) back to the underlying 3-D grid to allow updating of the mobilities before the pressure equation can be solved to recompute the velocity field.

A Specific Implementation. An implementation of the streamline method can be characterized by (i) the procedure for tracing streamlines, (ii) the choice of one-dimensional solver, (iii) the strategy for spatial distribution of streamlines, and (iv) the algorithms for mapping solution values back and forth between streamlines and the underlying (pressure) grid. We now describe what we shall refer to as our "standard" or "original" streamline method.

In this work we only consider models with Cartesian geometry, and we therefore use a simple semi-analytical tracing procedure due to Pollock (1988). Given the entry point and constant normal velocities on faces of a grid-block, Pollock's algorithm computes the exit point and the incremental time-of-flight associated with transversing the grid-block by assuming linear velocity variation in each direction. This way, each streamline can be traced numerically on a block-by-block basis from injector to producer or vice versa, or alternatively from an arbitrary point in the reservoir and forward to the producer and backward to the injector. After the tracing, each streamline is given as the indices of the blocks the streamline traverses, the entry and exit points, and the incremental time-of-flights for each block. These increments form the blocks in the streamline grid $\{\Delta\tau_{sl,i}\}$ on which (5) will be solved.

To solve the one-dimensional problems we employ front-tracking (see, e.g., Holden and Risebro, 2002), which is also applied in a commercial streamline simulator (Bratvedt et al., 1993; Bradtvedt et al., 1994). The front-tracking method is unconditionally stable and can directly utilize the time-of-flight grid resulting from the streamline trace, which makes the method very efficient and devoid of numerical diffusion. In contrast, solvers based on a finite-volume formulation typically need to map the initial data to a more regular grid (Batycky, 1997; Thiele, 2005).

The initial values for the one-dimensional problems are obtained by picking up the piecewise constant values from the underlying (pressure) grid, i.e., the grid-to-streamline mapping is the simplest possible,

$$S_{sl,i} = S_i. \tag{6}$$

To map values from the streamlines back to the grid, we use volumetric averaging. Volumes are associated with streamlines by considering each streamline as the centreline, or more precisely, as a representation of the cross-section, of a streamtube with an associated constant volumetric flux $q_{sl} = \mathbf{u}(\zeta)A(\zeta)$. This gives the volume of the streamline as,

$$\begin{aligned} V_{sl} \equiv V_{st} &= \int_0^s \phi(\zeta)A(\zeta)\mathrm{d}\zeta \\ &= \int_0^s q_{sl}\frac{\phi(\zeta)}{\mathbf{u}(\zeta)}\mathrm{d}\zeta = q_{sl}\tau_{sl}. \end{aligned} \tag{7}$$

The volume of a streamline in grid-block i is then $V_{sl,i} = q_{sl}\Delta\tau_{sl,i}$, and the precise definition of the streamline-to-grid volumetric averaging is,

$$S_i = \frac{\sum_{sl} S_{sl,i}V_{sl,i}}{\sum_{sl} V_{sl,i}}. \tag{8}$$

We note that considering streamlines as fluid carriers also makes it natural to define production characteristics simply by summing the contributions from all streamlines connected to each well. For long time-steps, the fractional flow of water at a producer may vary significantly; hence we measure the accumulated production along each streamline and define the total water production during a time-step of size Δt by,

$$\mathrm{PRD}_{\Delta t} = \sum_{sl} q_{sl}\int_{\Delta t} f_{w,sl}(t)\ \mathrm{d}t. \tag{9}$$

The values of (8) and (9), and the accuracy with which these values approximate the true saturation values and production increments, depend on how fluxes are assigned to the streamlines/streamtubes, and this may again be related to the procedure for distributing streamlines in the reservoir. Here we generate equally spaced starting points on the faces of grid-blocks containing injection wells. The number of starting points on each face is proportional to the volumetric flux across the face, which enables us to consider the streamlines as carrying approximately equal amounts of fluids, i.e., $q_{sl} \approx C$ for some constant C. An advantage of this approach is that the sums in (8) and (9) can be computed incrementally as streamlines are traced (Batycky, 1997) without knowing the associated volumetric flux, thus allowing completely independent processing of streamlines.

For the volumetric mapping (8) to make sense, each grid-block should in principle be traversed by at least one streamline. In general, there will be a number of grid-blocks that are not traversed by any of the streamlines

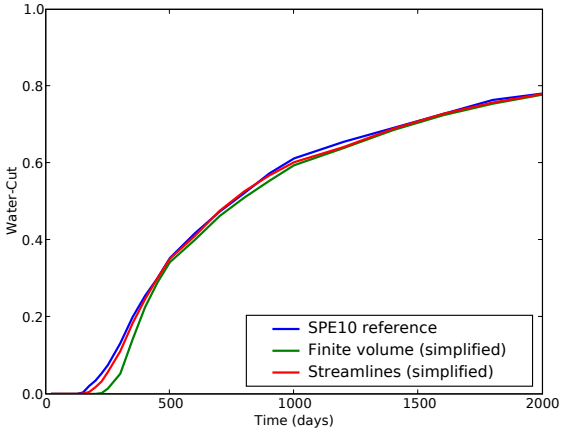


Fig. 1— Water-cuts for Producer 1 computed by a commercial streamline simulator on the original SPE10 model, along with our finite-volume and streamline solutions for the simplified problem (no compressibility and gravity).

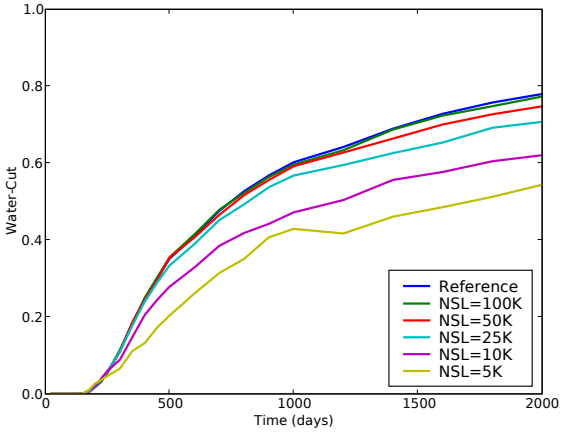


Fig. 2— Water-cuts for Producer 1 for various number of streamlines. (In the legend, 1K = 1 000.)

traced from the faces of injector-blocks. To make the streamlines cover all grid blocks, one can perform an additional tracing process where one picks a point inside one of the untraced blocks and traces a streamline from this point forward to a producer and backward to an injector, or one can follow Batycky (1997) and only trace streamlines back to a block which is already traversed by streamlines. The process is continued until there are no untraced blocks. Alternatively, one may simply ignore the untraced blocks, as these often are in regions that have a very small contribution to the production characteristics. To keep the amount of streamline tracing at a minimum we here employ the latter approach.

Mass-Balance Problems

For the particular streamline implementation described above, the overall accuracy will primarily depend on the number of streamlines used in the simulation. To illustrate the typical behaviour as the number of streamlines is reduced, we consider Model 2 from the 10$^{\text{th}}$ SPE Comparative Solution Project (Christie and Blunt, 2001), which is a large 3-D reservoir model consisting of $60 \times 220 \times 85$ grid-blocks, each of size 20ft $\times$ 10ft $\times$ 2ft. The model is a geostatistical realisation of a Brent sequence. The top 35 layers represent the Tarbert formation, which is a prograding near-shore environment. The lower 50 layers represent the Upper Ness formation, which is fluvial.

The model is produced using a five-spot pattern of vertical wells, where the central injector has an injection rate of 5 000 bbl/day (reservoir conditions), and the producers in each of the four corners of the model produce at 4 000 psi bottom hole pressure. As in the original model, we use quadratic relative permeability curves with $S_{\text{wc}} = S_{\text{or}} = 0.2$. The initial saturation is $S_0 \equiv S_{\text{wc}}$, and oil and water viscosities are $\mu_o = 3.0$ cP and $\mu_w = 0.3$ cP, respectively. For simplicity, we have neglected gravity and compressibility, since these have smaller impact on the production curves than the numerical diffusion inherent in any numerical scheme. This can be seen in **Fig. 1**, which compares a fine-grid reference solution from the SPE10 website (`http://www.spe.org/csp/`) with two fine-grid solutions for the simplified physical model; one computed by a first-order upstream finite-volume method, and the other computed by the "standard" streamline method introduced above with 600 000 streamlines. The simplified streamline solution will therefore be used as a reference solution in the rest of the paper. We note that we use the same time-steps as the reference solver: 25 steps with smaller step-sizes in the beginning of the simulation.

Figure 2 displays the water-cut in Producer 1 in simulations with various number of streamlines. The figure shows that the water production is underestimated when the number of streamlines is too small. Since the correct total amount of injected water is distributed among streamlines at the injecting end of each streamline, there must effectively be a loss of mass in the method. We can quantify this loss by, e.g., computing the relative mass-balance error for water in each time-step,

$$\epsilon_{\Delta t} = \frac{\text{INJ}_{\Delta t} - \text{PRD}_{\Delta t} + \text{FIP}_t - \text{FIP}_{t+\Delta t}}{\text{INJ}_{\Delta t}}, \qquad (10)$$

which is equivalent to the volume-balance error, since we have assumed incompressibility. **Figure 3** shows that the errors increase rapidly in the beginning of the simulation and decay slowly as the corresponding water-cut curves increase.

Spatial Errors in Streamline Discretizations. To explain the origin of the observed mass-balance errors we follow Jimenez et al. (2005): Using the bi-streamfunctions (Bear, 1972) for which,

$$\mathbf{u} = \nabla\psi \times \nabla\chi, \qquad (11)$$

we can define an alternative curvlinear coordinate system $(\tau,\ \phi,\ \chi)$ for three-dimensional space where the velocity $\mathbf{u}$, and hence the τ coordinate curves, i.e., the streamlines, will be orthogonal to the ψ and χ coordinate curves. It is this orthogonality relation (11) that is responsible for

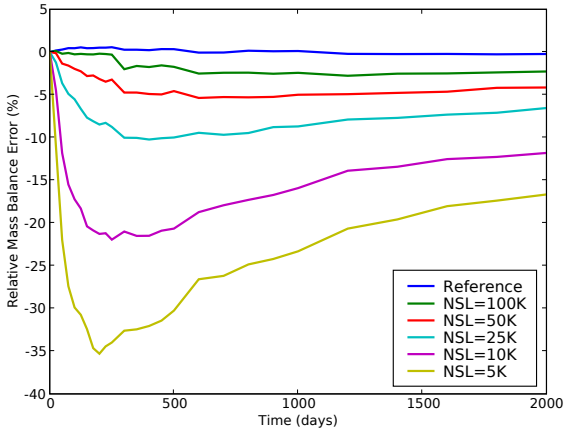


Fig. 3— Relative mass-balance errors in the streamline method for various number of streamlines.

the particularly simple form of the saturation equation (5) along streamlines. The discretization along streamlines is defined by the streamline grids obtained from the tracing algorithm, while the transversal discretization is determined by the partition of the volume into streamtubes, or in our case, the distribution of streamlines and the association of fluxes to streamlines.

The pore volume of this discretization will generally not match the pore volume of the original grid, which will lead to mass-balance errors when mapping saturation between the streamlines and the pressure grid. From (7) we have that the streamline pore volume is given by,

$$\mathrm{V}_{sl} = \sum_{sl} q_{sl}\tau_{sl}, \qquad (12)$$

but both q_{sl} and τ_{sl} are subject to approximation errors. As noted by Matringe and Gerritsen (2004), the simple semi-analytical streamline tracing approach gives errors, even if given analytical fluxes on the grid-block faces of Cartesian grids, since the velocity field is approximated by a piecewise bilinear function. Assigning equal fluxes to all streamlines is also slightly inaccurate since the fluxes actually represent the velocity integral across the cross-section of the associated streamtube. Even if the velocity is considered to be constant on injector-block faces, errors are introduced because the initial cross-section areas of the streamtubes will not be equal unless the number of starting points on each face is a perfect square. However, as the number of streamlines is reduced, the primary source of errors may be the assumption that the time-of-flight along a streamline is a good approximation to the average time-of-flight over cross-sections of the associated streamtube. To illustrate, **Fig. 4** shows the time-of-flight values for numerous points on the cross-section of two grid-blocks from the fluvial formation of the SPE10 model discussed above. Here, the variation in τ is actually of the same magnitude as the values themselves, hence the aforementioned assumption may yield very inaccurate streamline volumes.

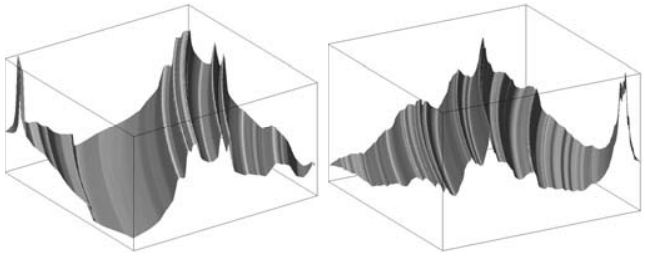


Fig. 4— Time-of-flight in two different grid blocks of Layer 76 in the SPE10 model sampled in 200×200 evenly distributed points inside each block.

Improving the Mass Balance

The presence of mass-balance errors in the streamline solutions is a well-known problem, and improving the accuracy and mass-balance properties of streamline methods is an active area of research. However, much of this research seems to be geared toward problems with complicated geometry and/or complicated physics. For instance, streamline tracing on corner-point geometry has been investigated by Jimenez et al. (2005) and Hægland et al. (2006), while Gerritsen et al. (2005; and related works) studied issues such as streamline distribution and more accurate mapping algorithms in the context of gas injection simulations. The latter works represent a fundamental change of the standard streamline approach, where streamlines are no longer viewed as fluid carriers, and saturations are mapped to the underlying grid using a statistical regression technique (kriging). This gives a large degree of freedom in distributing streamlines in the reservoir, but involves the solution of linear systems for the kriging weights, which may dominate the computation time for problems such as water-flooding, where the computation of the one-dimensional solutions is extremely efficient. Also, the natural way of estimating well production (9) is no longer applicable unless a volumetric flux is associated with each streamline.

Within the framework of regarding streamlines as fluid carriers, (12) shows that there are really only two parameters we can play with to improve the mass-balance properties of streamline methods, namely the streamline fluxes, q_{sl}, and the streamline time-of-flights, τ_{sl}. The exact properties of these are functions of the particular choices made in the streamline method implementation, and as noted in the previous section, both parameters may contain large errors for the specific implementation considered here. The close match between the reference streamline simulation and the finite-volume solver (**Fig. 1**) implies that the streamline tracing is sufficiently accurate for this problem, although we note that more accurate tracing on Cartesian grids is possible, for instance by utilizing velocity fields computed with higher-order mixed finite-element methods (Matringe et al., 2006).

The association of fluxes with streamlines is within the present framework related to the distribution of streamlines, since we assume equal flux for all streamlines. To bring the streamline fluxes closer to the actual velocity integral over the streamtube cross-section, we may consider scaling q_{sl} according to the interpolated velocity at the starting point and the cross-section area of the associated

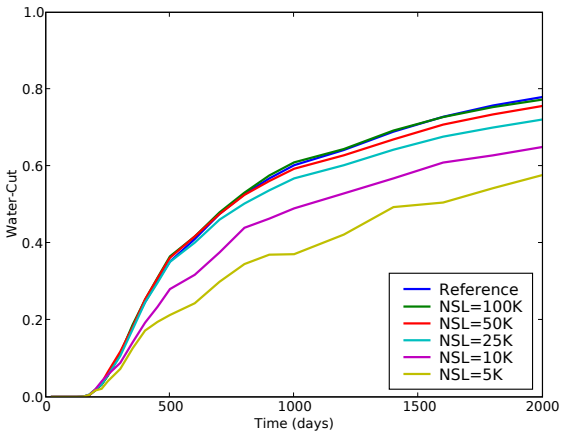


Fig. 5— Producer 1 water-cut for various number of streamlines when starting streamlines from both injectors and producers, and scaling the streamline fluxes according to the perpendicular bisection areas.

streamtube (Ponting, 1998; Pallister and Ponting, 2000). Lifting the restriction of equal streamline fluxes also makes it possible to apply other streamline distribution schemes. For instance, in situations where there is a large variation in total fluid rates between different producers, it may be beneficial to start streamlines also on the faces of well-blocks containing producers to ensure that sufficient accuracy is achieved even for wells with small rates.

To study the influence of these factors, we perform simulations where the starting points of streamlines are generated on the block-faces of both injectors and producers. For each face of grid-blocks containing an injector, the total volumetric flux is divided among the streamlines penetrating the face, with weights given by the areas of a perpendicular bisection of the block face. We have chosen to ignore the velocity variation over the faces since Ponting (1998) found that this only had a minor effect. **Figure 5** shows that this alternative approach to streamline distribution and flux computation is not significantly better than the original approach for the SPE10 case, although a slight improvement is detectable.

The results above indicate that the primary source of error, as the number of streamlines is reduced, is indeed that the time-of-flight along streamlines is not an accurate representation of the average time-of-flight over cross-sections of the associated streamtubes. Increasing the number of streamlines decreases the streamtube cross-sections and hence reduces these errors. However, considering the very large time-of-flight variation shown in **Fig. 4**, it appears that a large number of streamlines is necessary to obtain accurate streamline volumes and thereby low mass-balance errors. On the other hand, if we insist on keeping the number of streamlines low, we can use the fact that mass should be conserved, and correct the computed time-of-flight values to enforce the mass-balance constraint. We will have exact conservation of mass if the streamline volume matches the true pore volume, i.e., $\sum_{sl} V_{sl,i} = V_i$, in every grid-block touched by streamlines. In this case the mappings between streamlines and the pressure grid preserve mass. Indeed, for the grid-to-streamline mapping (6) we have,

$$\begin{aligned} V_{w,grid} = \sum_i V_i S_i &= \sum_i \Big(\sum_{sl} V_{sl,i}\Big) S_i \\ &= \sum_{sl} \sum_i V_{sl,i} S_{sl,i} = V_{w,sl}, \end{aligned} \tag{13}$$

and similarly for the streamline-to-grid mapping (8),

$$\begin{aligned} V_{w,sl} = \sum_{sl} \sum_i V_{sl,i} S_{sl,i} &= \sum_i \sum_{sl} \frac{V_i}{\sum_{sl} V_{sl,i}} V_{sl,i} S_{sl,i} \\ &= \sum_i V_i S_i = V_{w,grid}, \end{aligned} \tag{14}$$

where $V_{w,grid}$ and $V_{w,sl}$ are the total volumes of water on the the pressure and streamline grids, respectively. Since the streamline flux is constant along the streamline path, our only option for ensuring $\sum_{sl} V_{sl,i} = V_i$ is to modify the local time-of-flight values, $\Delta\tau_{sl,i}$. Specifically, prior to solving the one-dimensional saturation equation (5) along streamlines, we propose to scale the time-of-flight values in block i by a factor $\alpha_i = V_i / \sum_{sl} V_{sl,i}$. This means that streamlines can no longer be processed independently of each other, and we need to store streamlines in memory, or alternatively perform the complete tracing procedure twice; once to compute the values of α_i, and then a second time for the solution of the one-dimensional problems. The memory requirement for storing streamlines is usually lower than what is necessary for the solution of the pressure equation (1), hence the former approach is preferable since tracing is an expensive process.

Scaling the time-of-flight values amounts to locally stretching or shrinking the grid on which the one-dimensional saturation equation (5) is solved. Unless the errors in the streamline tracing are large, the grid obtained by the tracing is the correct one, and the modifications will result in less accurate one-dimensional solutions. In other words, by choosing to enforce mass conservation, we locally introduce errors, but as we demonstrate below, the global properties of the resulting solutions are better. However, we must take special care not to ruin important characteristics of the correct one-dimensional solutions. A quantity of particular interest is the breakthrough-time for producers, and to make sure this is estimated correctly, we only apply the time-of-flight scaling along streamlines where breakthrough has occurred.

Figures 6 and **7** show water-cuts for Producer 1 and and relative mass-balance errors for various number of streamlines when using the modified streamline approach. Mass-balance errors are still large initially since the time-of-flight scaling is only applied after breakthrough, but the errors decrease rapidly. The improvement on the water-cut curves is significant, to say the least, with as few as 5 000 streamlines giving acceptable results. We can quantify the error in a water-cut curve $w(t)$ by,

$$\delta(w) = \|w - w^{\text{ref}}\|_2 / \|w^{\text{ref}}\|_2, \tag{15}$$

and **Table 1** shows that the results are similar also for the other three producers. For completeness, **Table 1** also

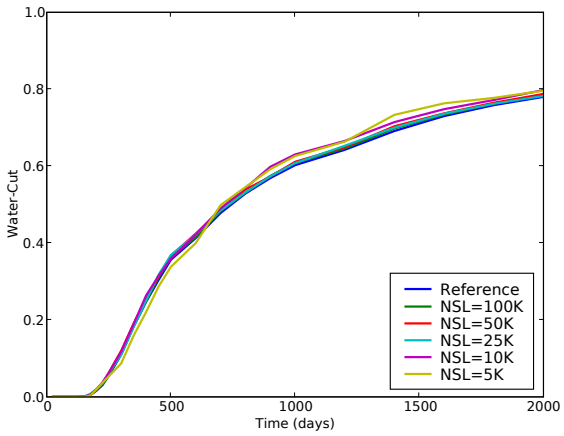


Fig. 6— Water-cuts for Producer 1 for various number of streamlines when using the modified streamline method.

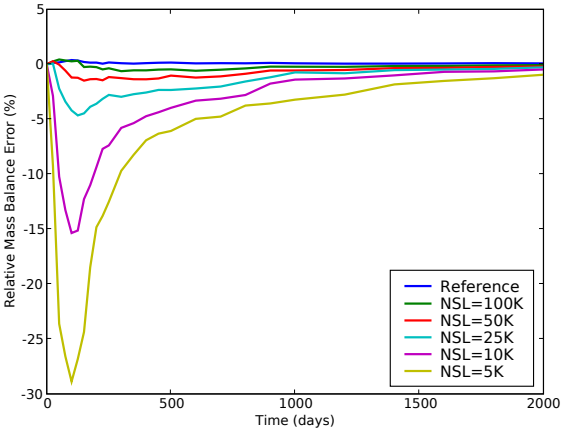


Fig. 7— Relative mass-balance errors in the streamline method for various number of streamlines when using the modified streamline method.

shows the corresponding results for the standard streamline approach, where we have started streamlines in both injectors and producers and used the perpendicular bisection approach above to assign fluxes to streamlines, as we previously showed that this gives slightly better results.

Since the time-of-flight scaling introduces local errors in the one-dimensional saturation solutions, we may ask if the global saturation solutions now are less accurate than for the original approach. We therefore compute saturation errors in the porosity-weighted L^1-norm,

$$\delta(S) = \|\phi(S - S^{\text{ref}})\|_1 / \|\phi S^{\text{ref}}\|_1, \qquad (16)$$

for each time-step, and the average through the simulation is displayed in **Table 1**. The results show that the modified streamline method gives slightly more accurate saturation fields than the original method for the same number of streamlines, but generally does not allow a significant reduction in the number of streamlines if one is to retain a certain accuracy.

If one, on the other hand, is mainly interested in accurate production curves, **Table 1** shows that the number of streamlines can be significantly reduced. For instance, if we allow a 5% water-cut error as measured by (15), we see that we may need 50 000 streamlines in the original approach, whereas 5 000 streamlines is sufficient when using the modified method. This yields a significant speedup for the transport part of the simulation, since the computation time associated with transport in theory scales linearly with the number of streamlines. The timing results in **Table 1** show that the actual scaling is not truly linear as the number of streamlines becomes very small. However, this is to be expected since our simulator is optimized for relatively large numbers of streamlines and otherwise negligible overhead associated with streamline distribution, flux computations, and saturation mappings may become significant when using a small number of streamlines. Still, we see that going from 5 000 to 50 000 streamlines gives at least five times speedup for the transport step. Finally, we note that as the number of streamlines is reduced, the total simulation time is dominated by the solution of the pressure equation (1). To obtain a more substantial speedup for the overall simulation, the modified streamline method should be considered in combination with approximate pressure solution techniques, e.g., with a multiscale method as discussed by Aarnes et al. (2005).

A Simpler Example

Since our proposed modification to the streamline method introduces local errors along streamlines, we might suspect that the approach only represents an improvement for difficult models where the original streamline method gives very large errors, and that it might yield significantly less accurate results for simpler models. In this section we therefore apply the modified and original streamline methods to a small model with homogeneous permeability and porosity data. In particular, we use a $32 \times 32 \times 8$ model with grid-block aspect ratio $1 : 1 : 0.1$, with wells placed in a five-spot pattern and the four producers producing at equal bottom-hole pressures. We assume quadratic relative permeability curves with zero residual oil and water saturations, and perform simulations for three different values of the end-point mobility ratio $M_{\text{end}} = \mu_o/\mu_w$, corresponding to unit mobility ratio ($M_{\text{end}} = 1$), favourable displacement ($M_{\text{end}} = 0.1$) and high-mobility ratio displacement ($M_{\text{end}} = 10$), respectively. The dimensionless simulation time was 2.0 PVI, and for all three scenarios we verified that the chosen number of time-steps was sufficient for stability of the sequential time-stepping scheme.

Tables 2 – 4 show the water-cut errors for both streamline approaches with various number of streamlines in each of the three scenarios. The results are indecisive and we cannot conclude that the modified streamline method significantly improves the accuracy of the production curves for this homogeneous model. On the other hand, the time-of-flight scaling does generally not cause the new approach to perform worse than the original method, either. This implies that the modified streamline approach can be safely applied also for simple datasets.

Adaptive Streamline Coverage The results in **Tables 2 – 4** also show that there is a large accuracy difference, for both methods, between the three displacement scenarios. In particular, **Table 3** shows that a larger number of streamlines is required to obtain accurate results for favourable displacement conditions. The large errors observed when using few streamlines for this piston-like displacement are caused by insufficient streamline coverage, since we do not ensure that all grid-blocks are traversed by streamlines. This leads to errors in the computed pressure and velocity fields, thus shifting the predicted time of breakthrough. For scenarios with high mobility-ratios, the pressure/velocity solutions are less sensitive to errors in the underlying saturation field, because the saturation variation is generally much smoother.

To alleviate the accuracy problems for favourable displacement conditions, we could trace streamlines through every grid-block, using, e.g., the approach of Batycky (1997). However, many grid-blocks will typically be located in low-flow regions that do not significantly affect the solution. We therefore propose an adaptive approach to streamline coverage, where we only demand that a given fraction β of the pore volume should be traversed by streamlines. Before the tracing starts, grid-blocks are sorted in descending order by absolute velocity $|\mathbf{u}|$, and we trace back from untouched blocks in sorted order until the given pore-volume target has been met. We also ensure that each well is properly covered by starting a specified number of streamlines from each well, with the distribution of streamlines on well-block faces given according to the fluxes, as before.

In **Tables 5 – 7** we have displayed the average number of streamlines and water-cut errors when applying this adaptive approach in combination with the modified streamline method for the three different displacement scenarios. Initially we trace 100 streamlines from each well, which is why the minimum number of streamlines is 500. Compared with **Tables 2 – 4** we see that tracing streamlines adaptively based on flow velocity gives more accurate results using fewer streamlines. As expected, the optimal value of β depends on the displacement conditions, with piston-like displacement requiring a larger fraction of the pore volume to be covered. On the other hand, the adaptivity has barely a significant effect for the scenario with high-mobility ratio, where we actually could have used even fewer streamlines. This helps explain why we obtained accurate results using extremely few streamlines for the SPE10 model above.

Application Example: History Matching

The results and discussion above clearly indicate that the modified streamline method is most suitable for applications where one is primarily interested in accurate production responses rather than accurate predictions of the dynamic distribution of fluids. Examples of such applications are history matching and ranking of multiple equiprobable geostatistical models.

In the following we consider history-matching of a high-resolution geomodel using a generalized travel-time inversion method (Vasco et al., 1999; He et al., 2002). The inversion method consists of four major steps that are repeated until a satisfactory match in production data is obtained: (i) Multiscale-streamline simulation to compute production responses at the wells. (ii) Quantification of the mismatch between observed and computed production responses via a generalized travel time, and computation of an optimal time shift that systematically shifts the computed production responses towards the observed data. (iii) Computation of streamline-based analytic sensitivities of water-cut data with respect to permeability. (iv) Updating of grid-block permeability values to match the production history via inverse modelling (minimization of a misfit functional). More details of the inversion procedure are given in (Stenerud et al., 2007).

We consider a synthetic geomodel given by a uniform Cartesian grid with $256 \times 128 \times 32$ cells, where each cell has dimensions 10m×10m×2m. A total number of 32 vertical injectors and 69 vertical producers are included in the simulation model. The injectors inject water at a constant total volumetric reservoir rate of 1609 bbl/day, and each producer produces fluids at a constant reservoir volume rate fulfilling the total voidage rate. The flow model assumes quadratic relative permeability curves with an end-point mobility ratio of $M_{\text{end}} = 5$.

The production history to be matched consists of 2475 days of water-cut data from the 69 producers, obtained by simulation on a reference geomodel having log-normally distributed permeability with mean 2.2 mD and values in the interval [0.017,79.5] mD; see **Fig. 8**. An initial permeability model was generated by assuming the permeability to be known in each well block and using sequential Gaussian simulation to generate multiple realizations.

The inversion algorithm converged in six iterations, after which the misfit in time-shift and amplitude (see Stenerud et al., 2007) were reduced to 7.8% and 53.6% of their initial values, respectively. In the inversion, we use the original streamline method with 500 000 streamlines and 15 uniform pressure steps of 165 days for each forward simulation. To solve the pressure equation we use an approximate, but highly efficient multiscale method (Aarnes and Lie, 2004). The total time for the whole inversion was 1 hour and 27 minutes on a Linux workstation with a 2.4 GHz Intel Core 2 Duo processor with 4Mb cache and 3 Gb memory. Using the modified streamline method we were able to reduce the number of streamlines to 50 000, and thereby reduce the total time for inversion to 39 minutes without reducing significantly the quality of the obtained history match. In fact, the time-shift and amplitude residuals were reduced to 7.6% and 48.7% of their original values, respectively.

Concluding Remarks

In this paper we have introduced a modified streamline method which greatly reduces the mass-balance errors when simulating large and complex reservoir models using few streamlines. Improved mass-conservation properties are achieved by locally scaling the time-of-flight grids, on which the one-dimensional transport equations are solved, to enforce mass conservation in the mappings between the Eulerian pressure grid and the Lagrangian streamlines. As a consequence, we are able to obtain accurate production curves on a million grid-block reservoir model with five

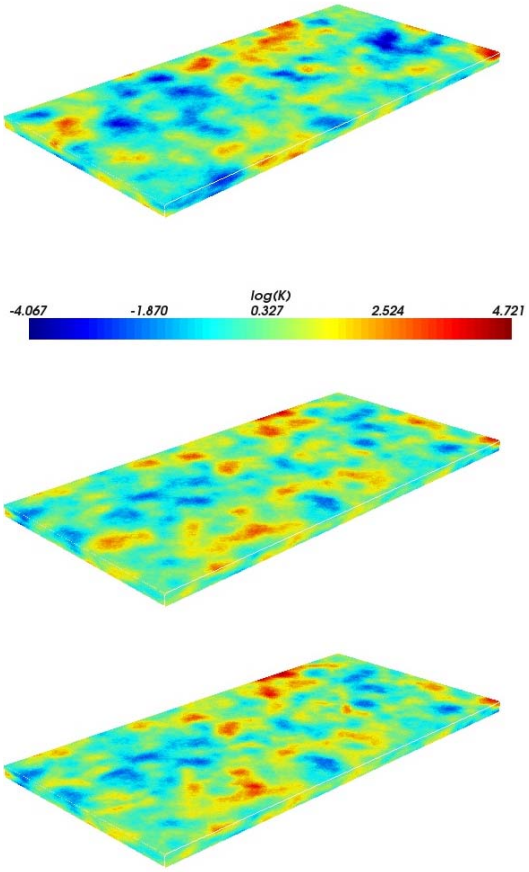


Fig. 8— Geomodels for the 3D history-matching case: reference model (top) initial model (middle), and final match (bottom).

wells using only 5 000 streamlines, and the total simulation time is less than ten minutes using a standard desktop computer.

We verified that the modified approach is applicable also to simple models by showing that its performance was similar to that of the original method on a homogeneous dataset. We also demonstrated that favourable, piston-like displacement might be challenging to simulate using few streamlines, and proposed an adaptive approach to streamline coverage based on tracing streamlines from untouched cells in high-flow regions until a given fraction of the pore volume has been traversed.

We have not considered improvements in the algorithms for mapping saturations between streamlines and the pressure grid, and the modified approach does therefore not give significantly better saturation solutions than the original method. This implies that the modified streamline method is best suited for applications that depend heavily on rapid estimation of production responses. As an example we demonstrated significant speed-up for a history-match of a million grid-block model with 32 injectors and 69 producers.

Finally, we remark that our modified method probably has an even larger potential for non-Cartesian grids, where Pollock's method for analytical tracing of streamlines introduces large errors, both in the time-of-flights and in the actual streamline paths; see Hægland et al. (2006) for further details.

Nomenclature

Roman letters

A	Area
C	Constant
f	Fractional flow
FIP	Volume of fluid (water) in place
$\mathbf{K}$	Permeability tensor
INJ	Injected volume (water)
k_r	Relative permeability
M_{end}	End-point mobility ratio
p	Pressure
PRD	Produced volume (water)
s	Distance along streamline
S	Saturation
S_{wc}	Connate water saturation
S_{or}	Residual oil saturation
t	Time
$\mathbf{u}$	Total Darcy velocity
V	Volume
w	Water-cut curve
q	Volumetric rate

Greek letters

$\delta(\cdot)$	Relative error
ϵ	Relative mass balance error
λ	Mobility
μ	Viscosity
ϕ	Porosity
ψ,χ	Bi-streamfunctions
τ	Time-of-flight
ζ	Space coordinate along streamline

Subscripts

i	Block number
j	Phase number
sl	Streamline number
st	Streamtube number
o, w	Oil and water phases
t, tot	Total
Δt	Time-step

Acknowledgements

The authors are grateful to V. R. Stenerud for providing the history-matching example, and to J. R. Natvig for providing the plot of time-of-flight within a grid cell. The authors gratefully acknowledge financial support from the Research Council of Norway; Kippe and Lie under grants number 152732/S30 and 158908/I30, and Hægland under grant number 173875/I30.

References

J. E. Aarnes and K.-A. Lie. Toward reservoir simulation on geological grid models. In *Proceedings of the 9th European Conference on the Mathematics of Oil Recovery*, Cannes, France, 2004. EAGE.

J. E. Aarnes, V. Kippe, and K.-A. Lie. Mixed multiscale finite elements and streamline methods for reservoir simulation of large geomodels. *Adv. Water Resour.*, 28(3):257–271, 2005.

R.P. Batycky. *A Three-Dimensional Two-Phase Field-Scale Streamline Simulator*. PhD thesis, Stanford University, Dept. of Petroleum Engineering, 1997.

J. Bear. *Dynamics of Fluids in Porous Media*. American Elsevier, New York, 1972.

F. Bradtvedt, K. Bratvedt, C.F. Buchholz, T. Gimse, H. Holden, L. Holden, R. Olufsen, and N.H. Risebro. Three-dimensional reservoir simulation based on front tracking. *North Sea Oil and Gas Reservoirs*, III:247–257, 1994.

F. Bratvedt, K. Bratvedt, C.F. Buchholz, T. Gimse, H. Holden, L. Holden, and N.H. Risebro. Frontline and frontsim: two full-scale, two-phase, black oil reservoir simulators based on front tracking. *Surv. Math. Ind.*, (3):185–215, 1993.

M.A. Christie and M.J. Blunt. Tenth SPE comparative solution project: A comparison of upscaling techniques. *SPE Reservoir Eval. Eng.*, 4(4):308–317, 2001. url: www.spe.org/csp.

M. Crane, F. Bratvedt, K. Bratvedt, P. Childs, and R. Olufsen. A fully compositional streamline simulator. In *SPE Annual Technical Conference and Exhibition, Dallas, TX, October 1 - 4, 2000*, SPE 63156, 2000.

A. Datta-Gupta and M. J. King. *Streamline Simulation: Theory and Practice*. SPE Textbook Series. to appear.

G. Di Donato and M.J. Blunt. Streamline-based dual-porosity simulation of reactive transport and flow in fractured reservoirs. *Water Resour. Res.*, 40(W04203), 2004. doi:10.1029/2003WR002772.

M.G. Gerritsen, K. Jessen, B.T. Mallison, and J. Lambers. A fully adaptive streamline framework for the challenging simulation of gas-injection processes. In *SPE Annual Technical Conference and Exhibition, Houston, TX, October 9 - 12, 2005*, SPE 97270, 2005.

H. Hægland, H.K. Dahle, G.T. Eigestad, K.-A. Lie, and I. Aavatsmark. Improved streamlines and time-of-flight for streamline simulation on irregular grids. *Advances in Water Resources*, 2006. doi:10.1016/j.advwatres.2006.09.002.

Z. He, A. Datta-Gupta, and S. Yoon. Streamline-based productoin data integration with gravity and changing field conditions. *SPE J.*, 7(4):423–436, 2002.

H. Holden and N.H. Risebro. *Front Tracking for Hyperbolic Conservation Laws*, volume 152 of *Applied Mathematical Sciences*. Springer, New York, 2002.

E. Jimenez, K. Sabir, A. Datta-Gupta, and M.J. King. Spatial error and convergence in streamline simulation. In *SPE Reservoir Simulation Symposium, Houston, TX, January 31-February 2, 2005*, SPE 92873, 2005.

M. J. King and A. Datta-Gupta. Streamline simulation: A current perspective. *In Situ*, 22(1):91–140, 1998.

S.F. Matringe and M.G. Gerritsen. On accurate tracing of streamlines. In *SPE Annual Technical Conference and Exhibition, Houston, TX, September 26 - 29, 2004*, SPE 89920, 2004.

S.F. Matringe, R. Juanes, and H.A. Tchelepi. Robust streamline tracing for the simulation of porous media flow on general triangular and quadrilateral grids. *JCP*, 2006. doi:10.1016/j.jcp.2006.07.004.

I.C. Pallister and D.K. Ponting. Asset optimization using multiple realizations and streamline simulation. In *SPE Asia Pacific Conference on Integrated Modelling for Asset Management, Yokohama, Japan, 25 - 26 April 2000*, SPE 59460, 2000.

D.W. Pollock. Semi-analytical computation of path lines for finite-difference models. *Ground Water*, 26(6):743–750, 1988.

D.K. Ponting. Hybrid streamline methods. In *SPE Asia Pacific Conference on Integrated Modelling for Asset Management, Kuala Lumpur, Malaysia, 23 - 24 March 1998*, SPE 39756, 1998.

V.R. Stenerud, V. Kippe, A. Datta-Gupta, and K.-A. Lie. Adaptive multiscale streamline simulation and inversion for high-resolution geomodels. In *SPE Reservoir Simulation Symposium, Houston, TX, February 26–28, 2007*, SPE 106228, 2007.

K. Stüben. *Algebraic Multigrid (AMG): An Introduction with Applications*. Academinc Press, 2000. Guest appendix in the book *Multigrid* by U. Trottenberg and C.W. Oosterlee and A. Schüller.

M.R. Thiele. Streamline simulation. In *8th International Forum on Reservoir Simulation*, Stresa / Lago Maggiore, Italy, 20-24 June 2005.

M.R. Thiele, R.P. Batycky, and M.J. Blunt. A streamline-based field-scale compositional reservoir simulator. In *SPE Annual Technical Conference and Exhibition, San Antonio, TX, October 5 - 8, 1997*, SPE 38889, 1997.

D.W. Vasco, S. Yoon, and A. Datta-Gupta. Integrating dynamic data into high-resolution models using streamline-based analytic sensitivity coefficients. *SPE J.*, pages 389–399, 1999.

NSL	O/M	P1	P2	P3	P4
2000	O	5.07e-02	2.33e-02	3.11e-02	2.90e-02
	M	3.93e-02	2.52e-02	2.69e-02	2.89e-02
1750	O	4.31e-02	2.86e-02	3.30e-02	2.86e-02
	M	3.64e-02	2.18e-02	2.37e-02	2.23e-02
1500	O	5.08e-02	2.91e-02	4.59e-02	3.21e-02
	M	5.11e-02	3.92e-02	3.15e-02	3.82e-02
1250	O	6.66e-02	5.21e-02	6.25e-02	5.49e-02
	M	6.30e-02	3.51e-02	4.55e-02	3.92e-02
1000	O	5.56e-02	4.33e-02	5.78e-02	3.97e-02
	M	4.73e-02	3.76e-02	5.00e-02	4.09e-02
750	O	9.25e-02	8.73e-02	9.09e-02	9.17e-02
	M	7.37e-02	7.03e-02	5.47e-02	7.30e-02
500	O	1.44e-01	1.12e-01	1.06e-01	1.15e-01
	M	1.19e-01	9.82e-02	9.64e-02	1.10e-01
250	O	1.68e-01	2.15e-01	2.43e-01	2.19e-01
	M	1.49e-01	1.61e-01	1.77e-01	1.59e-01

Table 2— Water-cut errors, $\delta(w)$, on the homogeneous model for the original (O) and modified (M) versions of the streamline method, when the end-point mobility ratio $M_{\text{end}} = 1$.

NSL	Method	P1	P2	P3	P4	$\delta(S)$	T_{sl} (s)	T_{tot} (s)
100 000	Original	8.91e-03	6.24e-03	2.44e-03	2.99e-03	2.75e-02	508.92	974.94
	Modified	9.86e-03	4.61e-03	1.97e-03	3.67e-03	2.83e-02	508.20	979.03
50 000	Original	2.53e-02	1.72e-02	6.42e-03	9.38e-03	4.00e-02	266.48	728.42
	Modified	1.66e-02	7.88e-03	3.72e-03	7.03e-03	3.81e-02	265.87	727.79
25 000	Original	6.49e-02	4.85e-02	1.74e-02	2.28e-02	5.89e-02	147.36	608.46
	Modified	1.43e-02	1.47e-02	8.12e-03	7.12e-03	5.27e-02	146.23	613.00
10 000	Original	1.78e-01	1.29e-01	5.53e-02	7.30e-02	9.54e-02	75.65	541.17
	Modified	3.26e-02	1.94e-02	1.56e-02	1.38e-02	8.06e-02	75.33	545.09
5 000	Original	3.20e-01	2.30e-01	1.02e-01	1.30e-01	1.29e-01	50.91	512.75
	Modified	4.25e-02	2.19e-02	1.86e-02	2.37e-02	1.12e-01	51.74	516.63

Table 1— Errors in water-cuts $\delta(w)$ for producers P1 to P4, saturation error $\delta(S)$, computational time for the streamline part of the simulation T_{sl}, and total computation time T_{tot} for the original and modified streamline methods on the SPE10 model for various number of streamlines (NSL).

NSL	O/M	P1	P2	P3	P4
2000	O	4.56e-02	9.18e-02	8.97e-02	9.46e-02
	M	4.43e-02	9.30e-02	8.97e-02	9.47e-02
1750	O	3.29e-02	6.53e-02	6.78e-02	5.17e-02
	M	3.07e-02	6.44e-02	7.09e-02	5.08e-02
1500	O	7.01e-02	1.20e-01	8.98e-02	1.05e-01
	M	5.27e-02	1.13e-01	8.96e-02	1.04e-01
1250	O	6.76e-02	1.13e-01	1.42e-01	8.19e-02
	M	4.60e-02	1.09e-01	1.42e-01	6.32e-02
1000	O	1.40e-01	1.62e-01	1.76e-01	1.67e-01
	M	1.39e-01	1.62e-01	1.76e-01	1.67e-01
750	O	3.40e-01	3.14e-01	3.39e-01	3.57e-01
	M	3.39e-01	3.15e-01	3.40e-01	3.57e-01
500	O	4.26e-01	4.50e-01	4.65e-01	4.69e-01
	M	4.25e-01	4.34e-01	4.64e-01	4.61e-01
250	O	7.79e-01	8.20e-01	8.44e-01	7.99e-01
	M	7.86e-01	8.17e-01	8.42e-01	8.00e-01

Table 3— Water-cut errors, $\delta(w)$, on the homogeneous model for the original (O) and modified (M) versions of the streamline method, when the end-point mobility ratio $M_{\text{end}} = 0.1$.

NSL	O/M	P1	P2	P3	P4
2000	O	2.55e-02	1.33e-02	5.36e-02	1.35e-02
	M	2.58e-02	2.45e-02	2.33e-02	8.44e-03
1750	O	2.89e-02	1.36e-02	6.15e-02	1.04e-02
	M	2.78e-02	1.94e-02	2.63e-02	8.13e-03
1500	O	3.37e-02	1.79e-02	4.43e-02	1.92e-02
	M	3.14e-02	1.00e-02	3.88e-02	9.23e-03
1250	O	3.93e-02	2.42e-02	4.93e-02	2.60e-02
	M	2.63e-02	2.61e-02	5.36e-02	2.72e-02
1000	O	5.33e-02	6.55e-02	3.33e-02	3.19e-02
	M	6.68e-02	2.29e-02	5.79e-02	4.14e-02
750	O	5.05e-02	3.52e-02	4.68e-02	4.27e-02
	M	5.84e-02	2.73e-02	5.09e-02	4.79e-02
500	O	6.97e-02	3.53e-02	5.19e-02	4.42e-02
	M	3.49e-02	4.35e-02	5.22e-02	3.14e-02
250	O	7.07e-02	9.37e-02	1.04e-01	1.01e-01
	M	7.42e-02	9.58e-02	1.20e-01	8.96e-02

Table 4— Water-cut errors, $\delta(w)$, on the homogeneous model for the original (O) and modified (M) versions of the streamline method, when the end-point mobility ratio $M_{\text{end}} = 10$.

β	NSL	P1	P2	P3	P4
1.0	875	3.54e-02	3.08e-02	2.50e-02	3.92e-02
0.9	714	3.86e-02	3.35e-02	3.14e-02	3.89e-02
0.8	576	4.74e-02	3.11e-02	2.28e-02	3.07e-02
0.7	500	6.88e-02	5.88e-02	7.22e-02	4.83e-02
0.6	500	6.83e-02	6.37e-02	7.03e-02	5.09e-02

Table 5— Average number of streamlines and water-cut errors, $\delta(w)$, on the homogeneous model for various values of β, when the end-point mobility ratio $M_{\text{end}} = 1$.

β	NSL	P1	P2	P3	P4
1.0	873	1.17e-02	7.52e-03	2.44e-02	1.37e-02
0.9	701	3.28e-02	2.95e-02	4.82e-02	2.17e-02
0.8	560	2.40e-01	2.31e-01	2.72e-01	2.43e-01
0.7	500	3.34e-01	3.85e-01	3.99e-01	3.90e-01
0.6	500	3.60e-01	3.78e-01	3.98e-01	3.86e-01

Table 6— Average number of streamlines and water-cut errors, $\delta(w)$, on the homogeneous model for various values of β, when the end-point mobility ratio $M_{\text{end}} = 0.1$.

β	NSL	P1	P2	P3	P4
1.0	873	3.45e-02	2.21e-02	2.26e-02	2.04e-02
0.9	722	3.42e-02	2.39e-02	2.41e-02	2.79e-02
0.8	616	2.69e-02	2.44e-02	3.35e-02	2.77e-02
0.7	519	2.19e-02	2.50e-02	5.94e-02	2.43e-02
0.6	500	2.39e-02	3.60e-02	6.80e-02	3.49e-02

Table 7— Average number of streamlines and water-cut errors, $\delta(w)$, on the homogeneous model for various values of β, when the end-point mobility ratio $M_{\text{end}} = 10$.