Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

Stian Falkfjell Rosvoldaune

Tracer-Based Two-Phase Upscaling

Master's thesis in Reservoir Engineering
Supervisor: Carl Fredrik Berg
June 2019

NTNU
Norwegian University of
Science and Technology

Stian Falkfjell Rosvoldaune

Tracer-Based Two-Phase Upscaling

Master's thesis in Reservoir Engineering
Supervisor: Carl Fredrik Berg
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Geoscience and Petroleum

NTNU
Norwegian University of
Science and Technology

SUMMARY

Accurate upscaling of subsurface flow is an important topic in the oil and gas industry due to the computational power requirements of performing simulations on complex geo-cellular models. The aim of an upscaling procedure is to remove some of the spatial detail of the complex models by replacing grid cells with blocks of homogeneous properties. The upscaling procedures are often classified based on the parameter that is upscaled and which type of information that enters the procedure. The main focus of this study will be on the upscaling of two-phase flow parameters, specifically the relative permeability.

One of the most common techniques for upscaling two-phase flow is steady-state upscaling. In steady-state upscaling, time-dependency is neglected such that a set of simplified equations are applicable. The most computationally demanding part of steady-state upscaling is finding the saturation distribution at steady-state which is used to calculate the upscaled relative permeability. To that end, this study introduces a new method referred to as tracer-based upscaling for finding the steady-state saturation distribution, which is based on converting saturation distributions from more computationally effective tracer simulations to actual saturation distributions. The conversion is performed by using what will be referred to as tracer functions, which are functions generated based on data from numerical experiments.

The use of tracer simulations to predict the saturation distribution is shown to provide equal accuracy or errors within a few percent with respect to an existing steady-state two-phase upscaling procedure and reference solutions from the original simulation model. The major attribute of the tracer-based upscaling is that this accuracy is obtained at a significantly lower computational cost than the existing steady-state two-phase upscaling procedures.

It is also shown that the tracer function is not model specific through the fact that accurate results are obtained by using functions generated from other models in the one used here. The fact that functions from other models gave accurate results indicates that the accuracy of the tracer-based upscaled results depend more on the shape of the tracer function than which model the function is based on. This is exemplified through the fact that the function's shape changes with time, and that more accurate results would be obtained if time-dependency was introduced to the function. Further investigations on how the shape of the function affects the results and if other models give different results are therefore the main recommendation on further work.

SAMMENDRAG

Nøyaktig oppskalering av strømning i reservoar er et viktig tema i olje og gass industrien på grunn av de store beregningskostnadene assosiert med å utføre simuleringer på komplekse reservoarmodeller. Målet med en oppskalering er å fjerne noen av de romlige detaljene til de komplekse modellene ved å erstatte celler med blokker som har homogene egenskaper. Oppskalering klassifiseres ofte basert på parameteren som blir oppskalert og hvilken type informasjon som brukes i oppskaleringen. Hovedfokuset i denne studien er oppskalering av parametere i tofasestrøm, nærmere bestemt relativ permeabilitet.

En av de vanligste oppskaleringsteknikkene for tofasestrøm er oppskalering ved stabil tilstand. Ved stabil tilstand neglisjeres tidsavhengighet slik at et sett med forenklede ligninger er anvendbare. Den mest beregningskrevende delen av oppskalering ved stabil tilstand er å finne metningsfordelingen som brukes til å finne den oppskalerte relative permeabiliteten. På bakgrunn av dette blir en ny metode for å finne metningsfordelingen ved stabil tilstand introdusert i denne studien. Metoden refereres til som sporpartikkelbasert oppskalering og baseres på å konvertere metningsfordelinger fra mer beregningseffektive sporpartikkelsimuleringer til faktiske metningsfordelinger. Konverteringen gjøres ved å bruke det som vil bli referert til som sporpartikkelfunksjoner, som er funksjoner som er generert av data fra numeriske eksperimenter.

Det vil bli vist at bruken av sporpartikkelsimuleringer til å finne metningsfordelingen ved stabil tilstand gir lik nøyaktighet eller feil innen noen få prosent sammenlignet med en eksisterende tofase oppskaleringsteknikk ved stabil tilstand og referanseløsninger fra den opprinnelige simuleringsmodellen. Hovedegenskapen til den sporpartikkelbaserte oppskaleringen er at denne nøyaktigheten oppnås med en mye lavere beregningskostnad enn for de eksisterende tofase oppskaleringsteknikkene ved stabil tilstand.

Det vises også at sporpartikkelfunksjonen ikke er modellspesifikk siden nøyaktige resultater oppnås ved å bruke funksjoner som er generert basert på andre modeller i modellen som benyttes her. Det faktum at funksjoner fra andre modeller gir nøyaktige resultater indikerer at nøyaktigheten av de oppskalerte resultatene fra den sporpartikkelbaserte oppskaleringen avhenger mer av formen på funksjonen enn hvilken modell funksjonen er basert på. Dette eksemplifiseres ved at funksjonens form endrer seg med tid og at mer nøyaktige resultater kan forventes dersom tidsavhengighet introduseres i funksjonen. Undersøkelser av hvordan funksjonens form påvirker resultatene fra den sporpartikkelbaserte oppskaleringen og om andre modeller gir forskjellige resultater er derfor hovedanbefalingen for videre arbeid.

ACKNOWLEDGEMENTS

This thesis is the final effort of my 5-years master's degree in Petroleum Geosciences and Engineering at the Norwegian University of Science and Technology in Trondheim, Norway. I have had five great years and look to the future with excitement.

I would especially like to offer my sincerest gratitude to my supervisor Assoc Prof. Carl Fredrik Berg for his guidance, not just in the master's degree period but during my entire final year. His curiosity, dedication, and kindness have been deeply appreciated and important for the progress of my work. Having such a talented supervisor has been an honor. I would also like to thank Stein Krogstad at Sintef for helping me with MRST whenever it was needed, and my friend and classmate Ivar for proofreading my thesis.

I wish to thank my family for all their support during my five years as a student at NTNU, including my mother Stine, father Otto and brothers Emil and Frode. My girlfriend Sylvia deserves the greatest gratitude for her love, patience, and backing. My five years as a student would have been a lot harder had it not been for you.

I also wish to thank my parents in law, Harry and Laila. Last, but not least, I offer my gratitude to my grandparents Agnes and Arne, who have always been of great support to me both in academics and life in general.

Thank you!

Stian Falkfjell Rosvoldaune
Trondheim, June 2019

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF CODE

ABBREVIATIONS & NOMENCLATURE

Abbreviations

AGMG	=	Aggregation-based algebraic multigrid
ALG	=	Adaptive local-global
CPU	=	Central processing unit
MPFA	=	Multi-point flux approximaion
MRST	=	The MATLAB Reservoir Simulation Toolbox
NCS	=	Norwegian continental shelf
PV	=	Pore volume
PVI	=	Pore volumes injected
REV	=	Representative elementary volume
SPE	=	Society of Petroleum Engineers
TOF	=	Time of flight
TPFA	=	Two-point flux approximation

Nomenclature

A	=	area
C	=	tracer concentration
c	=	compressibility
d	=	difference
F	=	fractional flow function
f	=	function
g	=	gravitational acceleration
h	=	thickness
J	=	Leverett J-function
k	=	permeability
$\mathbf{k}$	=	permeability tensor
k_p	=	effecive permeability of phase p
k_{rp}	=	relative permeability of phase p
L	=	system length
m	=	mass flow
N	=	number
n	=	normal
p	=	pressure
p_c	=	capillary pressure
p_p	=	pressure of phase p
q	=	flow rate
r	=	radius
S_p	=	saturation of phase p
s	=	streamline
T	=	temperature
T	=	transmissibility
t	=	time
$\mathbf{u}$	=	Darcy velocity
$\mathbf{u}_p$	=	Darcy velocity of phase p
V	=	volume
WI	=	well index
X	=	dataset
z	=	vertical coordinate
β	=	convergence criterion
χ	=	bi-stream function
δ	=	change
ϵ	=	threshold
λ	=	mobility
λ_p	=	mobility of phase p
μ	=	viscosity
μ_p	=	viscosity of phase p
Ω	=	control volume
ω	=	power averaging exponent
ϕ	=	porosity
ψ	=	bi-stream function
ρ	=	density
ρ_p	=	density of phase p
σ	=	interfacial tension
τ	=	time of flight
θ	=	contact angle

Subscripts

B	=	coarse block
b	=	bulk/backward
f	=	fluid/forward
i, j, k	=	block index
ir	=	irreducible
L	=	left
o	=	oil
p	=	pressure
Q	=	rate
R	=	right
r	=	residual saturation
t	=	total
w	=	water
x, y, x	=	coordinate direction
ξ	=	direction (x,y,x)

Superscripts

A	=	absolute
c	=	coarse scale
w	=	well
$\bar{\ }$	=	average
$\tilde{\ }$	=	per unit volume
$\vec{\ }$	=	vector
ν	=	iteration step
$*$	=	upscaled

CHAPTER 1

INTRODUCTION

In recent decades, a massive increase in computational power has enabled the oil and gas industry to utilize more complex methods for subsurface characterization. As described by Branets et al. (2008) typical fine-scale geo-cellular models consist of 10^6-10^8 cells, while the fine-scale simulation models are in the order of 10^5-10^6 cells.

Although it is possible to perform simulations directly on the fine-scale simulation models, it is seldom done due to the computational power requirements. It is therefore common to develop a new model which removes some of the spatial detail of the fine-scale geo-model through an upscaling procedure. The cells in this coarser model are referred to as blocks, as they are made up of several fine-scale cells. A region of fine-scale cells is replaced by a block with homogeneous properties which aims to capture the heterogeneity in the fine-scale region. An upscaling procedure is performed under the assumption that the results obtained from fine-scale simulations represent the actual conditions and flow behavior of the reservoir. The task is then to create a coarse model that reproduces the results from the fine-scale model (Christie, 2001).

Upscaling techniques can in general be classified as upscaling of single-phase parameters and upscaling of two-phase parameters. Single-phase upscaling considers the absolute permeability or interblock transmissibility and is the first step of any upscaling procedure. See e.g. Durlofsky (2005) for a detailed review on single-phase upscaling. Accurate single-phase results are important for the accuracy of the upscaled two-phase results as well. In fact, the upscaled single-phase parameters may give sufficient accuracy for the modeling of two-phase flow in some cases. In cases where e.g. the degree of coarsening is large or flow occurs in channels, an upscaling of the two-phase flow parameters are required.

In addition to the rock properties which are upscaled for single-phase flow, the upscaling of two-phase parameters involves rock-fluid properties, such as the relative permeability. The rock-fluid properties are time-dependent functions and are therefore more challenging to upscale than the single-phase flow parameters. This is due to the fact that time-dependent flow equations have to be solved at each time step which increases the computational costs significantly.

For both single- and two-phase upscaling, the upscaling techniques range from global to local methods based on the size of the flow problem. In local methods, the flow problems are solved within each coarse grid-block using assumed boundary conditions, see e.g. Durlofsky (2005). The imposed boundary conditions in a local method can have an important effect on the upscaled results, see e.g. Chen and Durlofsky (2006b) and Wallstrom et al. (2002b). In global methods, see e.g. Holden and Nielsen (2000), the flow problem is solved on the entire global domain in order to compute the upscaled parameters. By using global methods, any discrepancies in the results caused by the assumed boundary conditions in the local methods are eliminated. Global methods are however computationally demanding, and they are therefore often used only for single-phase upscaling. Another class of upscaling methods are the local-global upscaling methods, where local fine-scale flow problems are coupled with global coarse-scale flow problems, see e.g. Chen et al. (2003), Chen and Durlofsky (2006a), Wen et al. (2006), and Chen and Li (2009) for detailed discussions on local-global upscaling.

One of the most common techniques for upscaling two-phase flow is steady-state upscaling, see e.g. Odsæter et al. (2015), Stephen et al. (2001), and Ekrann and Aasen (2000). In steady-state upscaling the time-dependency is neglected, such that a set of simplified equations are applicable. The most demanding part of steady-state upscaling is finding the fine-scale saturation distribution at steady-state since multiple fine-scale flow simulations are needed. Under some conditions, the process of finding the saturation distribution may be simplified, but these conditions are not always present in a reservoir.

A new method for finding the saturation distribution will, therefore, be introduced. This method is based on using tracer simulations to predict the saturation distribution. Tracer simulations may be considered as simplified simulations where the complex two-phase fluids are replaced by simple fluids, see e.g. Ali et al. (2000), Møyner et al. (2015), Lie et al. (2015), Matringe and Gerritsen (2004), and Gupta and King (1995). The computational costs of the simulations to steady-state are then significantly reduced. The conversion from tracer saturation distribution to actual saturation distribution is performed by using a pre-defined function. The tracer-based upscaling is applicable for all reservoir conditions, and the accuracy of the results compared to existing methods with equal flow conditions depend only on the function which is used to convert the saturation distribution.

The software used to perform all simulations is The MATLAB Reservoir Simulation Toolbox (MRST), developed by the Computational Geosciences group in the Department of Mathematics and Cybernetics at SINTEF Digital (Lie, 2016). In order to limit the memory usage in MATLAB, an aggregation-based algebraic multigrid method (AGMG) is used as the linear solver in simulations (Notay, 2019). See Notay (2010), Napov and Notay (2012), and Notay (2012) for more detail on the AGMG method.

All simulations are performed on layers from the SPE10 model (Christie and Blunt, 2001), which has a size of $60 \times 220 \times 85$ cells (1.122×10^6 cells). The model is divided into two formations: the top 35 layers represent the Tarbert formation, and the remaining 50 layers represent the Upper Ness formation. The upper part of the model is a representation of a prograding nearshore environment, while the lower part is fluvial.

This thesis proceeds as follows: In Chapter 2 the governing equations on both the fine and coarse-scale are presented, along with descriptions of tracer flow and different upscaling procedures. Chapter 3 describes the upscaling procedures that was implemented in MRST, where one is the tracer-based upscaling technique. Results from simulations on the upscaled and fine-scale models are presented in Chapter 4, with the following discussion in Chapter 5. Conclusions are made in Chapter 6, followed by a recommendation of further work in Chapter 7.

CHAPTER 2 BACKGROUND

The relevant background theory for the upscaling procedures used in this thesis is presented in this chapter. The chapter starts by introducing the governing fine-scale equations for both single- and two-phase flow, and how these are made applicable for coarse grids. The concept of tracer simulations are then introduced, and the chapter is concluded by reviewing single- and two-phase upscaling procedures. Note that some simplifications are made in order to keep the level of detail at a basic level. Relevant papers are cited to provide additional information where simplifications are made or inadequate information is given.

2.1 Fine-scale equations

The equations describing single- and two-phase flow are considered in this section. Note that the equations presented in this section are only applicable to a fully resolved fine-scale model, which means that no upscaling of the equations has been performed.

2.1.1 Single-phase equations

The fundamental concept in deriving the single-phase flow equations is commonly known as Darcy's law. Using modern notation, Darcy's equation is

$$\mathbf{u} = -\frac{\mathbf{k}}{\mu}(\nabla p - \rho g \nabla z) \tag{2.1}$$

where $\mathbf{u}$ is the Darcy velocity through the porous medium, $\mathbf{k}$ is the permeability field, μ is the viscosity, ρ is the density of the fluid, g is the gravitational acceleration, p is the fluid pressure and z is the vertical coordinate.

Darcy's equation is derived by averaging fluxes inside a representative elementary volume (REV). A REV is defined as the smallest volume which may be considered representative for the whole volume considered. The equations and derivations presented next will also assume the existence of REVs, and consider control volumes like the one shown in **Fig. 2.1**.

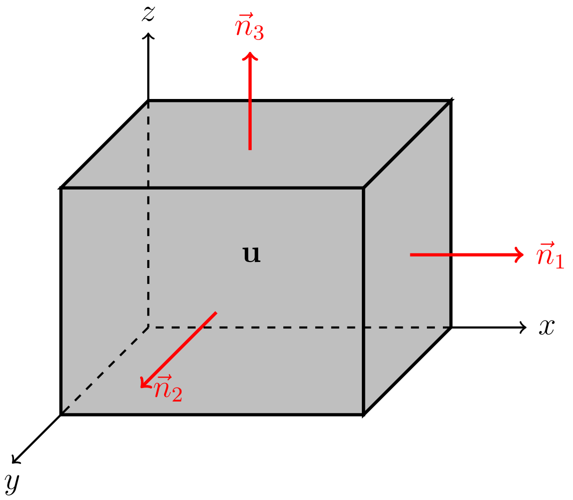


Figure 2.1: Illustration of a control volume Ω on which the principle of conservation is applied.

The fundamental law of mass conservation states that the total accumulation of mass inside a control volume must be equal to the net flux over the boundaries of the volume. Then, by letting Ω denote the control volume and $\partial\Omega$ denote the boundary of the control volume, conservation of mass gives

$$\frac{\partial}{\partial t}\int_{\Omega}\phi\rho d\vec{x} + \int_{\partial\Omega}\rho\mathbf{u}\cdot\vec{n}ds = \int_{\Omega}\rho\widetilde{q}d\vec{x} \tag{2.2}$$

where t is time, ϕ is the rock porosity, $\vec{n}$ is the normal at the boundary and $\widetilde{q}$ is a source/sink term. Rewriting equation (2.2) as

$$\int_{\Omega}\left[\frac{\partial}{\partial t}\phi\rho + \nabla(\rho\mathbf{u})\right]d\vec{x} = \int_{\Omega}\rho\widetilde{q}d\vec{x} \tag{2.3}$$

gives

$$\frac{\partial}{\partial t}(\rho\phi) + \nabla\cdot(\rho\mathbf{u}) + \widetilde{m} = 0 \tag{2.4}$$

where $\widetilde{m} = \rho\widetilde{q}$. Equation (2.4) is the continuity equation and by substituting Darcy's equation (2.1) in, the pressure equation is obtained as

$$\frac{\partial}{\partial t}(\rho\phi) - \nabla\cdot\left(\frac{\rho\mathbf{k}}{\mu}[\nabla p - \rho g\nabla z]\right) + \widetilde{m} = 0 \tag{2.5}$$

The pressure equation may be further modified by using the pressure dependency of the fluid density ρ and porosity ϕ. The pressure dependency of the density is derived based on the change in fluid volume

$$\frac{dV}{V} = \frac{1}{V}\left(\frac{\partial V}{\partial p}\right)_T dp + \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p dT \tag{2.6}$$

where V is the volume and T is the temperature. The subscripts p and T denotes that the changes takes place under constant pressure and temperature respectively. Since the mass flow $m = \rho V$ is constant, equation (2.6) may be written as

$$\frac{d\rho}{\rho} = \frac{1}{\rho}\left(\frac{\partial\rho}{\partial p}\right)_T dp + \frac{1}{\rho}\left(\frac{\partial\rho}{\partial T}\right)_p dT \tag{2.7}$$

Since the density changes slowly in most subsurface systems, heat conduction will keep the temperature constant (Lie, 2016), which reduces equation (2.7) to

$$\frac{d\rho}{\rho} = \frac{1}{\rho}\left(\frac{\partial \rho}{\partial p}\right)_T dp = c_f dp \tag{2.8}$$

The fluid compressibility c_f may thus be expressed as

$$c_f = \frac{1}{\rho}\frac{d\rho}{dp} \tag{2.9}$$

The pressure dependency of the porosity is found in a similar manner from the rock compressibility definition:

$$c_r = \frac{1}{\phi}\frac{d\phi}{dp} \tag{2.10}$$

Substituting these dependencies into the pressure equation gives

$$c_t \rho\phi \frac{\partial p}{\partial t} - \nabla \cdot \left(\frac{\rho \mathbf{k}}{\mu}[\nabla p - \rho g \nabla z]\right) + \widetilde{m} = 0 \tag{2.11}$$

where $c_t = c_r + c_f$ is the total compressibility. If the flow is considered incompressible, equation (2.11) is reduced to the elliptic equation

$$\nabla \cdot \left(\frac{\mathbf{k}}{\mu}\nabla[p - \rho g z]\right) = \widetilde{q} \tag{2.12}$$

The equation may be simplified further by neglecting the effects of gravity.

In order to numerically solve the pressure equation on a simulation grid, a finite difference discretization is applied. In the case of a two-dimensional system with diagonal permeability tensors, a two-point flux approximation (TPFA) is appropriate. This gives a five-point stencil for the pressure equation

$$\begin{aligned}(T_x)_{i-1/2,j}(p_{i-1,j} - p_{i,j}) + (T_x)_{i+1/2,j}(p_{i+1,j} - p_{i,j})\\ + (T_y)_{i,j-1/2}(p_{i,j-1} - p_{i,j}) + (T_y)_{i,j+1/2}(p_{i,j+1} - p_{i,j}) = 0\end{aligned} \tag{2.13}$$

where $p_{i,j}$ is the pressure in grid block (i,j) and $(T_x)_{i\pm1/2,j}$ and $(T_y)_{i,j\pm1/2}$ are intercell transmissibilities in the x and y direction. The extension of equation (2.13) to three dimensions is straightforward.

The transmissibility is a numerical quantity that relates the flow rate to the pressure difference between two cells. The transmissibility between cells (i,j) and $(i+1,j)$ can be defined as

$$(T_x)_{i+1/2,j} = \frac{2(k_x)_{i+1/2,j}\Delta y_j h}{\mu(\Delta x_{i+1} + \Delta x_i)} \tag{2.14}$$

where h is the model thickness and Δx_i and Δy_i are the dimensions of cell (i,j). The interface permeability is calculated by the harmonic average as

$$(k_x)_{i+1/2,j} = \frac{(\Delta x_i + \Delta x_{i+1})(k_x)_{i,j}(k_x)_{i+1,j}}{\Delta x_{i+1}(k_x)_{i,j} + \Delta x_i (k_x)_{i+1,j}} \tag{2.15}$$

The transmissibilities and interface permeabilities in the y and z directions are found in a similar manner as above. If wells are present in the model, these are modeled by a well index. The well index relates well flow rate q^w to the difference between well pressure p^w and cell pressure $p_{i,j}$ such that

$$q^w = WI(p_{i,j} - p^w) \tag{2.16}$$

2.1.2 Two-phase equations

The derivation of the two-phase flow equations are similar to the single-phase case. For two-phase flow, however, a set of equations is needed for each phase. Darcy's equation is then expressed as

$$\mathbf{u}_p = -\frac{\mathbf{k}k_{rp}}{\mu_p}(\nabla p_p - \rho_p g \nabla z) \tag{2.17}$$

where the subscript p refers to the phase and k_r is the relative permeability. The continuity equation for each phase may be written as

$$\frac{\partial}{\partial t}(\phi \rho_p S_p) + \nabla \cdot (\rho_p \mathbf{u}) + \widetilde{m}_p = 0 \tag{2.18}$$

where S_p is the saturation of phase p. For the rest of this section, a simplified system of incompressible immiscible flow of oil and water will be considered. It is assumed that the densities does not vary in time or space, that $\partial\phi/\partial t = 0$, and that gravity is negligible.

The total Darcy velocity for this system is

$$\mathbf{u}_t = \mathbf{u}_w + \mathbf{u_o} = -\mathbf{k}\left(\frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o}\right) \cdot \nabla p \tag{2.19}$$

The water velocity $\mathbf{u}_w$ may be expressed by a fractional flow function as $\mathbf{u}_w = f(S_w)\mathbf{u}_t$ as described by Buckley and Leverett (1942). Substituting this into the continuity equation (2.18) gives the water saturation equation

$$\phi \frac{\partial S_w}{\partial t} + \nabla \cdot [\mathbf{u}_t f(S_w)] = -\widetilde{q}_w \tag{2.20}$$

The pressure equation for two-phase flow may be found in a similar manner. Under the previously states assumptions, the continuity equation is reduced to

$$\nabla \cdot \mathbf{u}_t = -\widetilde{q}_t \tag{2.21}$$

Using the expression for the total velocity then gives

$$\nabla \cdot [\mathbf{k}\lambda_t \cdot \nabla p] = \widetilde{q}_t \tag{2.22}$$

where the total mobility λ_t is defined as

$$\lambda_t = \lambda_w + \lambda_o = \frac{k_{rw}}{\mu_w} + \frac{k_{ro}}{\mu_o} \tag{2.23}$$

To accompany the water saturation equation (2.20) and the pressure equation (2.22), two more equations may be stated. The saturation of phase p is defined as the fraction of the pore volume which is occupied by phase p. In the case of single-phase flow, it is assumed that the entire pore volume is filled with the flowing fluid. In two-phase flow however, the assumption is that the pore volume is occupied by both the fluids, i.e.

$$\sum_p S_p = S_w + S_o = 1 \tag{2.24}$$

There will also be a difference in phase pressures in a two-phase system. This pressure difference is known as the capillary pressure, and it is defined as

$$p_c = p_o - p_w \tag{2.25}$$

where $p_c = p_c(x, S_w)$. The system of four equations thus have four unknowns: S_p and p_p, for $p = o, w$.

2.2 Coarse-scale equations

The equations presented in the previous section are only applicable to a fully resolved fine-scale model, which means that upscaling of parameters are needed for them to be valid on a coarse grid. In order to keep the upscaled equations as simple as possible, some assumptions are made. In the case of single-phase flow it is assumed that the flow is incompressible with no source terms in the equations, while for two-phase flow an immiscible and incompressible displacement of oil and water without source terms in the equations is considered.

2.2.1 Upscaling of single-phase equations

When an upscaling procedure is performed, a region of fine-scale cells is replaced by a block with homogeneous properties that aims to capture the heterogeneity in the underlying fine scale region. When equations with variable parameters are homogenized, the upscaled equations often have a different form than the original fine-scale equations (Chen et al., 2003). In some cases when there is a separation of scales, Saez et al. (1989) among others have demonstrated that the upscaled pressure equation is on the same form as the fine-scale pressure equation. The difference is that the permeability **k** is replaced by an effective permeability tensor **k***, such that the upscaled pressure equation is

$$\nabla \cdot (\mathbf{k}^* \cdot \nabla p^c) = 0 \tag{2.26}$$

where the superscript $*$ indicate upscaled coarse-scale quantities and the superscript c represents coarse-scale volume-averaged quantities. For a more detailed discussion on the validity of this approximation, see e.g. Durlofsky et al. (1997), Saez et al. (1989), and Durlofsky (1991).

Numerous methods of calculating the effective grid block permeability exist, where the simplest is the power averaging method. By assuming the effective grid block permeability and the fine-scale permeability are diagonal tensors, the upscaled permeability may be found as (Deutsch, 1989)

$$k_i^* = \left(\frac{1}{V_b^c} \int_{V_b} [k_i(\mathbf{y})]^{\omega_i} dV \right)^{1/\omega_i}, \qquad -1 \leq \omega \leq 1 \tag{2.27}$$

where k_i^* are the diagonal elements of the effective grid block permeability tensor, V_b^c is the coarse bulk volume, $k_i(\mathbf{y})$ are the diagonal elements of the fine-scale permeability tensor, and ω_i is the power averaging exponent.

Wen and Hernandez (1996) found that the averaging exponent is case specific and closely dependent on block shape, flow conditions, and heterogeneity. This means that in order to obtain the correct exponent, numerical experiments have to be performed. In a simple case of flow parallel to the layers in a model the exponent $\omega = 1$ which equals the arithmetic average, while for flow perpendicular to the layers $\omega = -1$ which corresponds to the harmonic average.

The power averaging method is favorable due to its simplicity and the fact that it is applicable to both two and three-dimensional grids. In more general cases, the upscaled permeability will be a full tensor such that the power averaging method is not applicable. In these cases, the TPFA presented in equation (2.13) has to be replaced by a multipoint flux approximation (MPFA). For discussions on full tensor upscaling of permeability and MPFA, see Efendiev (2000), Wen et al. (2000), Aavatsmark (2002), and Chen et al. (2007).

2.2.2 Upscaling of two-phase equations

As for the upscaling of the single-phase equations, the upscaled two-phase equations are also often taken to be on the same form as the fine-scale equations ((2.20) and (2.22)) such that

$$\nabla \cdot [\mathbf{k}^* \lambda_t^* \cdot \nabla p^c] = 0 \tag{2.28}$$

$$\phi^c \frac{\partial S_w^c}{\partial t} + \nabla \cdot [\mathbf{u}_t^c f^*(S_w^c)] = 0 \tag{2.29}$$

where

$$\mathbf{u}_t^c = -\lambda_t^* \mathbf{k}^* \cdot \nabla p^c \tag{2.30}$$

Similarly, as for the single-phase equations, the upscaled coarse-scale parameters have to be found through an upscaling procedure such that the upscaled equations produce results as close to the fine-scale equations as possible.

In some cases it is sufficient to only perform an upscaling of the single-phase parameters ($\mathbf{k}^*$), which means that the fine-scale relative permeability functions are retained in the coarse-scale model. The problem of such models is that they do not account for the transport effects. Retaining the fine-scale relative permeability functions is applicable when the heterogeneity of the fine-scale permeability field is small, which often is the case for small degrees of coarsening or nonuniform grids where the heterogeneity has been minimized inside the individual coarse grid blocks. Durlofsky (1998) gives a theoretical justification for neglecting the upscaling of relative permeability in cases of low heterogeneity.

As previously mentioned, the superscript c denotes volume-averaged coarse-scale variables. A way of obtaining exact coarse-scale equations is through volume-averaging of the fine-scale equations themselves instead of the variables (Chen and Li, 2009). The volume averaged equations are

$$\nabla \cdot \overline{[\lambda_t \mathbf{k} \cdot \nabla p]} = 0 \tag{2.31}$$

$$\overline{\phi} \frac{\partial \overline{S_w}}{\partial t} + \nabla \cdot \overline{[\mathbf{u}_t f(S_w)]} = 0 \tag{2.32}$$

where the overline represents the volume averaging. The volume averaged equations will be valid when the averages are taken over rectangular grid blocks with a constant porosity. See Durlofsky (1998) for a detailed discussion on volume averaged equations and their relationship to the existing upscaling techniques.

Another more recent way of solving the two-phase equations on a coarse grid is the multiscale methods. The idea of these methods is to incorporate fine-scale information into the coarse-scale equations by solving local fine-scale flow problems. An upside of these methods is that they are almost independent of the geometry of the coarse grid since the local problems are solved only on the fine grid.

Due to the nature of the multiscale methods, they offer great flexibility in terms of the solution method for the local flow problems. The solution method for each local problem may be chosen independently from the other problems, which enables the use of higher-order methods in areas of interest which in turn gives higher accuracy. See Aarnes et al. (2007) for examples of multiscale methods and a more detailed discussion.

2.3 Tracer flow

In reservoir simulation, tracer flow may somewhat simplified be considered as adding neutral particles that do not alter the flow properties to the flow (Lie, 2016). There are numerous applications of tracer flow simulations, where some of which are reservoir characterization and flow diagnostics. The flow in reservoirs is mostly controlled by the heterogeneities in the reservoir, and it is therefore important to be able to resolve the flow paths through these heterogeneities (Ali et al., 2000). One way of achieving this is by matching tracer simulation results with measured tracer data between e.g. an injector-producer pair.

The main focus here will be on the use of tracers to compute and visualize diagnostics of the flow. Flow diagnostics are simplified methods that give quantitative information about the flow behavior in a reservoir through numerical experiments (Møyner et al., 2015). Computing flow-diagnostics is usually a lot faster than traditional full-scale reservoir simulations, and may thus be used as a way of e.g. comparing different realizations, optimizing and visualizing.

To model tracer flow, the continuity equation is used such that

$$\frac{\partial(\phi C)}{\partial t} + \nabla \cdot (\mathbf{u}C) = \widetilde{q}_C \tag{2.33}$$

where C is the tracer concentration and q_C is a source term. Using this equation will give solution methods equal to that of single-phase flow described in Section 2.1.1, with the tracer concentration as the primary unknown. A more simplified method of performing tracer simulations is to compute the stationary solution. The stationary solution is found by simulating the evolution of artificial and non-diffusive tracers with a concentration that does not change with fluid compression of expansion. In other words, a time step in a simulation is "locked" and tracer is injected. By letting the tracer concentration be one in a fluid source, the solution that is approached at steady-state conditions may then be found by solving the equations

$$\mathbf{u} \cdot \nabla C_i = 0, \quad C_i|_{inflow} = 1 \tag{2.34}$$

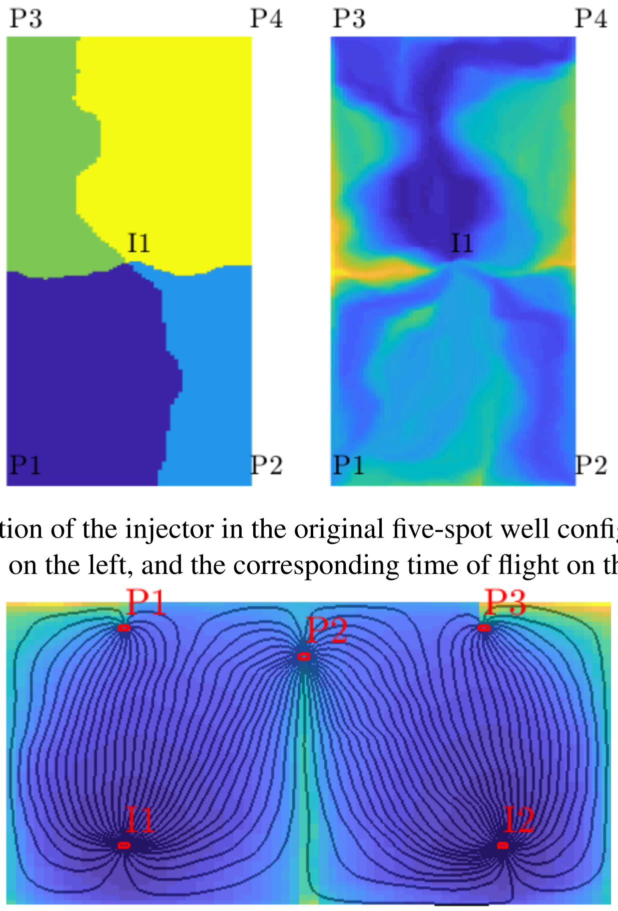
$$-\mathbf{u} \cdot \nabla C_p = 0, \quad C_p|_{outflow} = 1 \tag{2.35}$$

The tracer concentrations found by solving these equations will give the portions of volumes that will be allocated to a given injector or producer. The left figure in **Fig. 2.2a** shows such a partitioning for each injector-producer pair in the SPE10 model. As previously mentioned, it is important to be able to resolve the flow paths through the reservoir when computing flow diagnostics. Using stationary tracer solutions do not give any indications on how the flow is influenced by the heterogeneity. To introduce time dependence to the stationary solution, a time of flight (TOF) coordinate may be used (Shahvali et al., 2015).

The TOF is the time it takes a tracer particle released at a point in the reservoir to reach another given point. TOF is usually associated with streamline methods as (Møyner et al., 2015)

$$\tau(r) = \int_0^r \frac{\phi(\vec{x}(s))}{\mid \mathbf{u}(\vec{x}(s)) \mid} ds \tag{2.36}$$

where τ is the time a tracer particle uses to travel a distance r along a streamline in the interstitial velocity field $\mathbf{u}/\phi$. Fig. 2.2b shows an example of streamlines between well pairs and the corresponding time of flight distribution.



(a) Stationary tracer partition of the injector in the original five-spot well configuration of Model 2 from Christie and Blunt (2001) on the left, and the corresponding time of flight on the right.

(b) Illustration of streamlines (black lines) overlaying the time of flight distribution.

Figure 2.2: Examples of the use of tracer simulations and TOF in flow diagnostics.

Applying the fundamental theorem of calculus along with a directional derivative to equation (2.36) gives

$$\frac{d\tau}{dr} = \frac{\phi}{\mid \mathbf{u} \mid} = \frac{\mathbf{u}}{\mid \mathbf{u} \mid} \cdot \nabla \tau \tag{2.37}$$

Both a forward and backward TOF may be found. The forward TOF is the time it takes a tracer particle to travel from an injector to a given point in the reservoir, while the backward TOF is the time it takes a tracer particle to travel from a given point to a producer. An example of a forward TOF is shown in the right figure of Fig. 2.2a. By using equation (2.37), transport equations similar to equations (2.34) and (2.35) may be stated for the time of flight

$$\mathbf{u} \cdot \nabla \tau_f = \phi, \quad \tau_f|_{inflow} = 0 \tag{2.38}$$

$$-\mathbf{u} \cdot \nabla \tau_r = \phi, \quad \tau_r|_{outflow} = 0 \tag{2.39}$$

Unlike the tracer flow results, the TOF will give an indication on how a multiphase displacement may evolve with fixed wells or boundary conditions and provide additional information about the flow field than the pressure and velocity alone will. An example of such is finding the breakthrough time of e.g. water from an injector based on the saturation profile obtained from a typical Buckley-Leverett displacement as illustrated in **Fig. 2.3**. Note that the figure is just for illustration purposes, such that the shape of the displacement may be different when the x-axis is shown as TOF instead of distance.

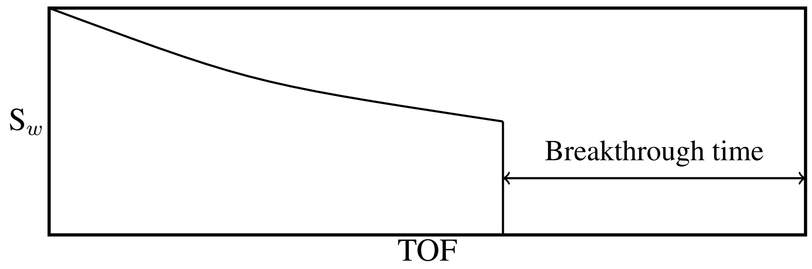


Figure 2.3: Illustration of the use of TOF to predict breakthrough time in a typical Buckley-Leverett displacement.

The TOF will also give indications of regions that are not of interest, regions with a high TOF are likely to remain unswept and are therefore targets for new production wells (Møyner et al., 2015). By combining the TOF and tracer distributions, it is also possible to compute well-allocation factors or flow-capacity and storage diagrams, which are a measure of the heterogeneity of the reservoir (Shahvali et al., 2015).

Notice that the only difference between the tracer and TOF equations are the boundary conditions and the right sides. The computation of the TOF and tracer concentration may thus be formulated as a single linear system with multiple right sides that may be solved by using a reordering technique that permutes the linear system of equations into a triangular or block-triangular form (Lie et al., 2015). See Natvig et al. (2007) or Natvig and Lie (2008) for more detail on this solution procedure.

Another way of simplifying the tracer transport equation (2.33) is to use streamlines and TOF to define a curvilinear and flow-based coordinate system in three dimensions (Lie, 2016). This is done by introducing the bi-stream functions ψ and χ such that $\mathbf{u} = \nabla\psi \times \nabla\chi$. The gradient operator in (τ, ψ, χ)-coordinates is then

$$\nabla = (\nabla\tau)\frac{\partial}{\partial\tau} + (\nabla\psi)\frac{\partial}{\partial\psi} + (\nabla\chi)\frac{\partial}{\partial\chi} \tag{2.40}$$

Since $\mathbf{u} \cdot \nabla\psi = 0$ and $\mathbf{u} \cdot \nabla\chi = 0$ (Chen et al., 2013), substitution into equation (2.38) gives

$$\mathbf{u} \cdot \nabla = (\mathbf{u} \cdot \nabla\tau)\frac{\partial}{\partial\tau} = \phi\frac{\partial}{\partial\tau} \tag{2.41}$$

For incompressible flow, equation (2.33) is reduced to

$$\frac{\partial C}{\partial t} + \frac{\partial C}{\partial\tau} = 0 \tag{2.42}$$

by using the coordinate transform. Equation (2.42) indicates that each streamline may be viewed as an isolated system. Gupta and King (1995) then suggests that if there is a concentration history $C_0(t)$ at the inflow, the solution will be a time-dependent boundary-value problem for the outflow concentration. For equation (2.42), the solution is $C(t) = C_0(t-\tau)$. See e.g. Aarnes et al. (2005), Matringe and Gerritsen (2004), and Thiele (2001) for more detail on streamlines and streamline simulation methods such as the one presented here.

The observant reader may have noticed that the two-phase pressure equation (2.22) is similar to the single-phase pressure equation (2.12). The equations will be equal if the two phases in equation (2.22) have identical properties, that is $k_{rw} = S_w$, $k_{ro} = S_o = 1 - S_w$ and $\mu_w = \mu_o$. The results from simulations will then equal those of a tracer simulation. Since the two fluids (oil and water) have equal properties the pressure solution is time independent and the transport equation can be solved for the entire simulation time at once.

2.4 Upscaling procedures

In order to solve the upscaled equations presented in Section 2.2, the parameters in the equations have to be found such that they are applicable to a coarse grid. An upscaling procedure is therefore necessary, and some of these procedures are presented in this section. The upscaling procedures are usually classified in terms of the parameter that is upscaled, and the type of information that enters the procedure.

For single-phase flow, the only parameters to be upscaled are the porosity and the permeability or transmissibility. The relative permeability has to be upscaled in addition to the single-phase parameters in the more general case of two-phase flow. The second type of classification differentiates between local and global upscaling procedures. An explanation of the differences between these will be given below.

2.4.1 Local and global upscaling procedures

Most upscaling techniques are based on an averaging procedure of parameters inside the region made up of a coarse grid block. These techniques are called local upscaling procedures since the information that enters the upscaling is solely obtained from within the grid block. **Fig. 2.4** shows examples of solution domains used in local upscaling procedures.

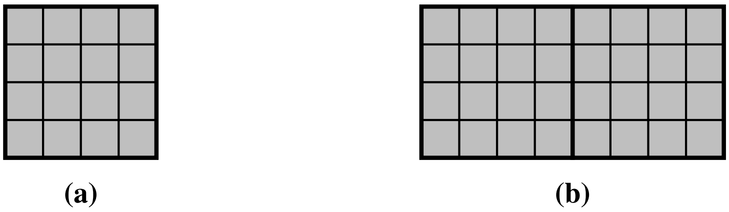


Figure 2.4: Illustration of local solution domains for (a) permeability upscaling and (b) transmissibility upscaling.

The thick lines in Fig. 2.4 represents a coarse grid block, while the thinner lines are the underlying fine-scale cells. The pressure and transport equation are solved on the subgrid making up the coarse block under some assumed boundary conditions. The obtained solutions are then used to find the upscaled parameters. Because the choice of boundary conditions will affect the outcome of the upscaling, there is always some uncertainty in specifying the boundary conditions.

The boundary conditions generally used in upscaling procedures may be divided into two categories: fixed and periodic boundary conditions. An example of fixed boundary conditions are the constant pressure - no flow conditions. A pressure difference is then introduced across the local domain and all outer faces but the inflow and outflow face are set as no-flow faces, i.e. no flow enters or leaves these faces. In the case of permeability upscaling, these boundary conditions are applicable when only the diagonal terms of the tensor are required.

When using periodic boundary conditions, the assumption is that e.g. the permeability field within a coarse block is repeated in all directions around the block (Pickup et al., 1994). Unlike the fixed boundary conditions, periodic conditions ensure that the upscaled permeability tensor is symmetric and positive definite. Durlofsky (1991) and Pickup et al. (1994) show that the use of periodic boundary conditions usually gives more robust upscaled models than the fixed boundary conditions.

Since the upscaling in local methods is based only on information obtained from within each coarse grid block, they do not account for the communication between neighboring blocks. The consequence of this is that the upscaled parameters may not reflect the global flow patterns in the reservoir. It is therefore desirable to include some of the global effects into the upscaling procedure. The use of extended-local upscaling (also called oversampling) is one of the ways to achieve this.

In extended-local upscaling, the local solution domain is enlarged with a border region of fine-scale cells as illustrated in **Fig. 2.5**. Wu et al. (2002) among others have demonstrated that the use of border regions do in fact improve the accuracy of the upscaled results. The size of the extended domain varies depending on the computational cost associated with the upscaling procedure and which parameter that is upscaled. If the global flow effects and heterogeneity are important, e.g. in the case of permeability upscaling, the local domain should be enlarged by the fine-scale cells equal to the neighboring coarse-scale grid blocks.

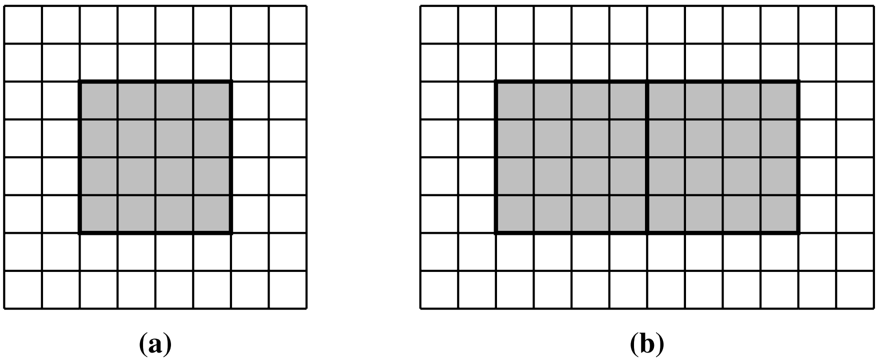


Figure 2.5: Illustration of extended-local solution domains for (a) permeability upscaling and (b) transmissibility upscaling.

A reasonable expectation when using border regions is that the influence of the choice of boundary conditions will be less significant compared to the purely local methods, and that results using different boundary conditions will tend to converge (Durlofsky, 2005). The solution method in an extended-local upscaling procedure is similar to that of purely local procedures. In the case of e.g. single-phase permeability upscaling the pressure equation is solved on the full extended domain, while the upscaled permeability is found by only using information from within the target coarse grid block.

As previously mentioned, the use of local upscaling procedures may cause the results to inaccurately capture the flow behavior of the model due to the lack of global flow information. Global upscaling procedures aim to account for these effects by solving flow problems on the fine-scale grid and using these solutions to find the parameters on the coarse-scale. This is illustrated in **Fig. 2.6**, with the target coarse blocks shaded in gray.

Holden and Nielsen (2000) motivates the use of global upscaling procedures in cases where the assumed boundary conditions used in local upscaling procedures are far from the actual boundary conditions in the reservoir. In such cases, the error in the estimates will be large. The global upscaling procedures will provide accurate results for a particular set of external influences but may lack robustness if these are changed (Durlofsky, 2005).

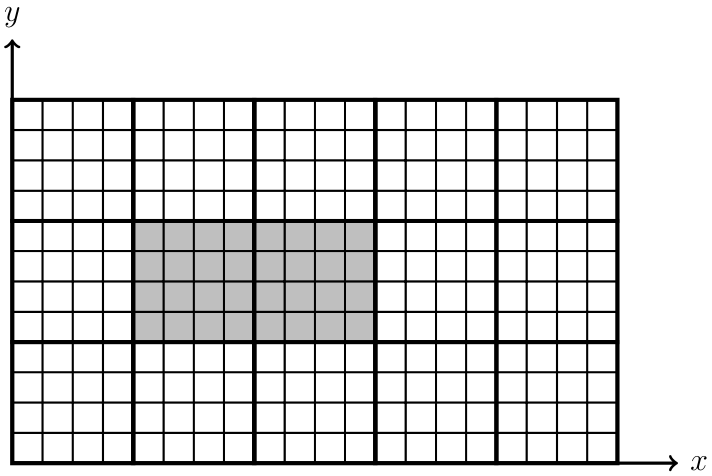


Figure 2.6: Illustration of a global solution domain.

A particular drawback of the global upscaling methods is the need for full fine-scale simulations. In the more general cases of two-phase flow, these are computationally demanding and time-consuming. In such cases the global methods may seem to contradict the purpose of upscaling, especially since iterations often are needed to get sufficient accuracy in the results.

A way of avoiding the full fine-scale simulations was proposed by Chen et al. (2003) through their local-global upscaling procedure. This method combines elements from both local and global upscaling. Rather than performing full fine-scale simulations, coarse-scale simulations are performed to find the boundary conditions to use in extended-local upscaling. A schematic showing a domain with coarse pressures, p^c, shown on the boundaries are displayed in **Fig. 2.7**.

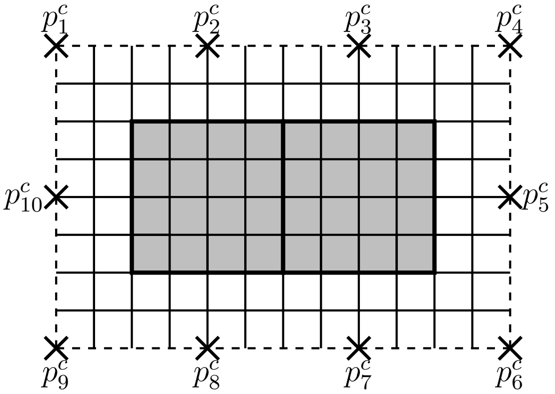


Figure 2.7: Schematic showing a domain for transmissibility upscaling for which the coupling described by Chen et al. (2003) is performed. The ×'s are coarse pressures.

Once the initial estimates are found by a purely local upscaling, a global coarse-scale pressure solution is found. The coarse pressures surrounding the target coarse block(s) are then interpolated to give the fine-scale boundary pressures for an extended-local problem. An extended-local upscaling procedure is then performed with the new boundary conditions. Since the initial estimates may give poor local boundary conditions, an iteration procedure is performed to ensure consistency between the local and global results. The convergence criteria for the iterations are based on minimizing changes in pressure and velocity between iteration steps such that self-consistency is obtained.

This approach was later generalized to account for changes in well configuration or global boundary conditions by Chen and Durlofsky (2006a), and extended to three dimensions by Wen et al. (2006). The approach presented by Chen and Durlofsky (2006a) also includes a thresholding procedure for which upscaled transmissibilities are only found for interfaces subject to a sufficient amount of flow, which they found reduced the occurrences of anomalous and negative transmissibilities which is often a problem in global upscaling.

The iterations may cause the local-global upscaling procedure to be time-consuming if the initial estimates are inaccurate, especially in a three-dimensional case. Wen et al. (2006) showed that the use of a purely local method for the initial estimates and use of reduced border regions in the repeated extended-local upscaling gave results close to those obtained by using "full" border regions. When full border regions are used, the border region includes the fine-scale cells which corresponds to the neighboring coarse-scale grid blocks of the target blocks.

Another way of avoiding full fine-scale simulations was proposed by Nielsen and Tveito (1998) with a weighted output least squares minimization technique. They found the upscaled permeability by solving an optimization problem where the difference between the fine- and coarse-scale velocity field is minimized, without performing fine-scale simulations. This is achieved by using a weighted L^2-norm in the output least squares formulation.

2.4.2 Two-phase upscaling

Compared to single-phase upscaling, two-phase upscaling is much more challenging to perform both with regards to methodology and computational cost. In addition to rock parameters like the porosity and permeability, two-phase upscaling includes rock-fluid properties like relative permeability which are time-dependent functions.

As discussed in the previous section, the choice of local boundary conditions has an important effect on the upscaled results. This is especially the case for two-phase upscaling since conditions for both pressure and saturation are needed. Due to the time-dependency of the saturation, both the boundary conditions and the non-local effects will evolve in space and time (Chen and Li, 2009). See Chen et al. (2013), Chen and Durlofsky (2006b), and Wallstrom et al. (2002b) for discussions and comparisons on different local boundary conditions for two-phase upscaling.

Most two-phase upscaling procedures are local due to the computational costs of a global procedure. The global procedures eliminate the need to assume the local boundary conditions however, which in most cases results in more accurate results. For most linear solvers, the computational cost is proportional to the number of unknowns (Holden and Nielsen, 2000). This means that for single-phase flow where only one global pressure solution is required, the global methods may be applied. For two-phase flow however, time-dependent flow equations have to be solved at each time step, which increases the computational costs significantly. This means that in practice, global upscaling of two-phase flow parameters is not feasible. Similarly as for single-phase flow, Chen and Li (2009) proposed a coupling between local and global upscaling to reduce the computational costs of upscaling of two-phase flow.

Aarnes et al. (2007) explains the difficulties of two-phase upscaling by considering the saturation equation in its simplest form

$$\phi\frac{\partial S_w}{\partial t} + \nabla \cdot f_w(S_w)\mathbf{u} = 0 \tag{2.43}$$

with the discretization

$$\overline{\phi}\frac{S_i^{n+1} - S_i^n}{\Delta t} + \sum_j \left[F_w(S_i, S_j) Q_{ij} \right] = 0 \tag{2.44}$$

where S_i^n is the average saturation in a grid block at time t_n, $\overline{\phi}$ is the mean porosity and Q_{ij} is the total flux across the interface between two grid blocks. The numerical fractional flow function F_w is given as

$$F_w(S_i, S_j) = \begin{cases} f_w(S_i), & \text{if } Q_{ij} \geq 0 \\ f_w(S_j), & \text{if } Q_{ij} < 0 \end{cases} \tag{2.45}$$

In order to couple the coarse- and fine-scale grid with this scheme, spatial fractional flow functions F_{ij} that represent the average flux across each coarse interface have to be found. These flow functions are often referred to as pseudo-functions, and they are dependent upon both the heterogeneous media, the saturation distribution within each coarse grid block, and well locations and/or global boundary conditions. The pseudo-functions are usually divided into equilibrium and dynamic pseudo-functions. The use of equilibrium pseudo-functions is similar to steady-state two-phase upscaling (Aarnes et al., 2007), which will be discussed later in this section.

The dynamic pseudo-functions aim to produce coarse-scale solutions equal to those of averaged fine-scale solutions. An example of a method which uses dynamic pseudo-functions was presented by Jacks et al. (1973), where upscaled relative permeability functions for each coarse interface is found by using a transmissibility-weighted average of the fine-scale relative permeabilities in the upstream cells. See Aarnes et al. (2007) for a more detailed discussion on the use of pseudo-functions in two-phase upscaling.

Steady-state upscaling is an alternative to the pseudo-functions. The essential assumption is steady-state, i.e. that the time derivative in the saturation equation is zero such that the steady-state saturation equation takes the form

$$\nabla \cdot \left[\frac{\mathbf{k} k_{rp}}{\mu_p} (\nabla p_i - \rho_i g z) \right] = 0 \tag{2.46}$$

The assumption of steady state do not hold in all areas of a reservoir, but in areas away from saturations fronts and in regions with low flow the saturations changes will be small and the assumption can be justified. Odsæter et al. (2015) divides steady-state upscaling into the following steps

1. Calculate the fine-scale saturation distribution, $S_w(\mathbf{x})$ at steady state.
2. Calculate the permeability distribution, $\mathbf{k}_p(\mathbf{x}, S_w) = k_{rp}(\mathbf{x}, S_w)\mathbf{k}(\mathbf{x})$ from input data.
3. Perform single-phase upscaling to calculate the upscaled permeability tensor $\mathbf{k}_p^*$

Following these three steps, the upscaled relative permeability is found as $\mathbf{k}_p^* \cdot \mathbf{k}^{-1}$ where $\mathbf{k}$ is the upscaled permeability tensor. The relative permeability is found in conjunction with the upscaled water saturation, which is simply found as a volume-weighted average of the fine-scale saturation. A detailed review of the steps shown above will be given in Chapter 3.

The most demanding part of steady-state upscaling is finding the fine-scale saturation distribution, since it requires fine-scale flow simulations to reach the steady-state. In some special cases, however, the process of finding the saturation distribution may be simplified. At steady-state, the system of two-phase equations may be written as

$$\mathbf{u} \cdot \nabla f_w + \nabla \cdot (\lambda_o f_w \nabla p_c) = 0 \tag{2.47}$$

where p_c is the capillary pressure. If gravity is included, equation (2.47) is balanced by three forces: viscous forces, capillary forces and gravitational forces. The process of simplifying the calculation of the saturation distribution is performed by looking at the limits of the balance between the forces.

In the capillary limit, one assumes that the capillary pressure is constant in time and space, which is a reasonable assumption in areas of low flow rates. In these areas, the capillary forces will dominate the viscus and gravitational forces such that these can be neglected. Equation (2.47) then reduces to

$$\nabla \cdot (\mathbf{k}\lambda_o f_w \nabla p_c) = 0 \tag{2.48}$$

Since the capillary pressure is a known constant in each grid block, the saturation distribution may be found by inverting the capillary pressure curve such that

$$p_c(\mathbf{x}, S_w) = P_c \tag{2.49}$$

where P_c is the constant capillary pressure. The capillary pressure curve is a strictly monotone function which means that solving equation (2.49) will give a unique solution. The capillary pressure function is often modelled by the Leverett J-function (Aarnes et al., 2007), which has the form

$$J(S_w) = \frac{p_c(S_w)\sqrt{\frac{\mathbf{k}}{\phi}}}{\sigma \cos(\theta)} \tag{2.50}$$

where σ is the interfacial tension and θ is the contact angle. The interfacial tension and contact angle are measured in laboratory experiments. Since the J-function is usually a known tabulated function of S_w, the fine-scale saturation distribution may be found as

$$S_w = J^{-1}\left(\frac{P_c\sqrt{\frac{\mathbf{k}}{\phi}}}{\sigma \cos(\theta)}\right) \tag{2.51}$$

The capillary limit method will also produce an upscaled capillary pressure function

$$\overline{p}_c(\overline{S}_w) = P_c \tag{2.52}$$

Remember that the assumption of capillary dominated flow holds for areas of low flow rates. For areas of high flow rates, the flow will be dominated by the viscous forces. At the viscous limit, one assumes that the capillary and gravitational forces are negligible such that equation (2.47) reduces to

$$\mathbf{u} \cdot \nabla f_w = 0 \tag{2.53}$$

Equation (2.53) indicates that the fractional flow is constant. With the additional assumption that the pressure gradient is equal for both phases and that the relative permeability of oil is nonzero, the ratio of the phase velocities is

$$\frac{u_w}{u_o} = \frac{-\frac{\mathbf{k}k_{rw}}{\mu_w}\nabla p}{-\frac{\mathbf{k}k_{ro}}{\mu_o}\nabla p} = \frac{k_{rw}\mu_o}{k_{ro}\mu_w} \tag{2.54}$$

The saturation distribution may then be found by solving the equations

$$S_w + S_o = 1, \quad k_{rw}(S_w, \mathbf{x}) = \left(\frac{u_w \mu_w}{u_o \mu_o}\right) k_{ro}(S_o, \mathbf{x}) \tag{2.55}$$

For more detail on the capillary and viscous limit solutions see e.g. Ekrann and Aasen (2000). A challenge of the steady-state methods presented above is to determine which method is most appropriate. On a small scale neglecting the viscous forces is a fair assumption, while on a large scale the contribution from capillary forces is negligible. This means that the balance of the forces is scale-dependent (Hilden and Berg, 2016). Stephen et al. (2001) proposed to normalize the contributions from each force with the sum of the forces and plot them in a diagram like the one shown in **Fig. 2.8** to help decide on which method to use.

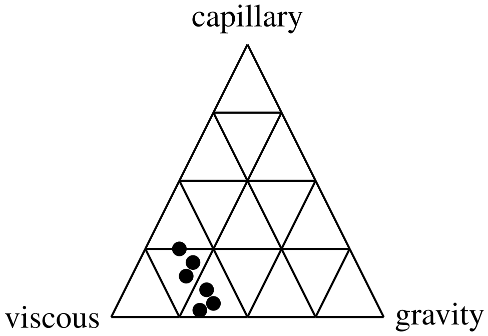


Figure 2.8: Example diagram from Stephen et al. (2001) showing the balance of forces in steady-state upscaling.

The balance may also be found by using laboratory experiments on core samples. By comparing observed rates from core-flooding experiments with results from numerical simulations, the impact of the different forces may be found. Care should be taken when performing the laboratory experiments however since the balance of forces is scale-dependent as previously discussed. Other methods include that of Odsæter et al. (2015), where a scale-dependent capillary number is used.

The discussion above has focused on the upscaling of relative permeability. It is also possible to find the upscaled mobility λ^* and fractional flow f^* for interfaces or grid blocks, and use them to derive the upscaled relative permeabilities as

$$k^*_{rw}(S^c) = \lambda^*(S^c) f^*(S^c) \mu_w \tag{2.56}$$

$$k^*_{ro}(S^c) = \lambda^*(S^c)[1 - f^*(S^c)] \mu_o \tag{2.57}$$

Also worth mentioning is that the amount of data that is required for a coarse model is very large since the two-phase flow functions usually vary for each coarse grid block. The requirements of handling of this data may not be manageable, which calls for some kind of grouping technique to reduce the number of upscaled two-phase flow functions (Chen et al., 2013).

As a final remark on two-phase upscaling, it is worth mentioning that there is still no best-practice. The pseudo-functions and steady-state methods described above may be successful for some models, but there is still the issue of finding the correct boundary conditions, achieving robustness and process independence, grouping relative permeabilities, etc. (Christie, 2001). The fact that different models and reservoir structures may require different upscaling methods implies that two-phase upscaling should ideally be linked to the gridding of the simulation models (Aarnes et al., 2007).

CHAPTER 3
METHODOLOGY

This chapter gives a detailed description of the work done in this thesis. The first sections give an overview of the single- and two-phase upscaling procedures that was used and introduces the use of tracers to find the saturation distribution in steady-state two-phase upscaling. The chapter is concluded by giving an overview of the simulation model.

3.1 Single-phase upscaling

Since the main focus of this work is to investigate the upscaling of two-phase flow parameters, it is important that the upscaled single-phase parameters are as accurate as possible. Two highly accurate single-phase flow upscaling procedures are therefore used and compared, where one is a global procedure and the other is an extended-local procedure.

3.1.1 Global transmissibility upscaling

The global transmissibility upscaling procedure used is the *upscaleTrans* function in MRST, whose outline is shown in **Fig. 3.1**. Global upscaling procedures may as previously mentioned be time-consuming due to the full fine-scale simulation, but for single-phase flow only one global pressure solution is required and the computational cost is therefore limited. Global single-phase upscaling should be used with care however since the computational cost in a global method does not scale linearly with the number of cells in the model.

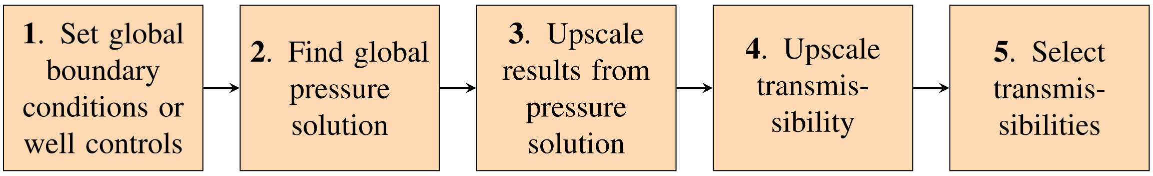

1. Set global boundary conditions or well controls
2. Find global pressure solution
3. Upscale results from pressure solution
4. Upscale transmissibility
5. Select transmissibilities

Figure 3.1: General outline of the *upscaleTrans* function from Lie (2016).

The first step in the global upscaling procedure is defining which global boundary conditions to use. If there are no wells in the model, a unit pressure drop along each axial direction is used. One case is made for each direction, such that a pressure solution is required for each case. If there are wells in the model, well cases rather than boundary condition cases are made. The well configuration is then considered linearly independent and one global pressure solution is found for each well. This is achieved by injecting 10 m^3/day through each well and producing through the others.

The global pressure solution in step 2 is found on the fine grid for the cases with the boundary conditions or well configurations described above. Note that incompressible flow with no gravitational effects is assumed. The pressure solution is found by using a highly efficient TPFA-scheme with minimum memory consumption found in MRST as *incompTPFA*.

Once the global fine-scale pressure solution is obtained it is upscaled, shown as step 3 in Fig. 3.1. The upscaled pressure is simply found as the volume-weighted average of the fine-scale solution

$$p_B^c = \frac{1}{V_b}\sum_{i=1}^{N_i} p_i V_i \tag{3.1}$$

where V_b is the total volume of coarse block B and p_i and V_i is the pressure and volume of a fine-scale cell $i \in B$. Note that using the cell volumes instead of the pore-volumes of the cells may give inaccurate upscaled pressures. An example that substantiates this is that there might be cells with zero porosity which do not contribute to the flow through a grid block. When the cell volumes are used, the contributions from these cells will be included. The flux between coarse block B and $B+1$ is found as

$$u_{B+1/2}^c = \sum_{i=1}^{N_f}(\mathbf{u}_i \cdot n) \tag{3.2}$$

where the sum is taken over the interfaces of the fine-scale cells along with the interface of blocks B and $B+1$. If wells are present in the model, upscaled well rates are also found by summing the rates in the fine-scale cells which makes up the well path in the coarse model in a similar way as in equation (3.2).

Once the upscaled pressures and flow rates are found, the upscaled transmissibilities may be found as (see Fig. 2.4b for an example of a domain for which the upscaled transmissibility is calculated)

$$T_{B+1/2}^* = \frac{Q_{B+1/2}}{p_B^c - p_{B+1}^c} \tag{3.3}$$

for all interfaces in the coarse model. If wells are present, the upscaled well index is found as

$$WI^* = \frac{q_w}{p_w - p_B^c} \tag{3.4}$$

where q_w is the well rate, p_w is the well pressure and p_B^c refers to the pressure in the well block. In step 5 transmissibilities which are not finite are set to $10^{-6} \cdot \min(T^*)$. The result is then a set of finite transmissibility data for each case of external influences. A decision is then made on which transmissibility to use. This is done by choosing the transmissibility of the case which has the largest flux across the interface in question, and the well index of the case which has the largest well flux. Note that if negative or anomalous transmissibilities occur, they are replaced by a transmissibility derived from an absolute permeability upscaling.

3.1.2 Adaptive local-global transmissibility upscaling

The presentation of the ALG transmissibility upscaling is a compressed version of the work done in the specialization project, where the ALG procedure by Wen et al. (2006) was extended to irregular coarse grids. ALG upscaling is as previously mentioned a way of incorporating global effects into a local upscaling through the procedure shown in **Fig. 3.2**.

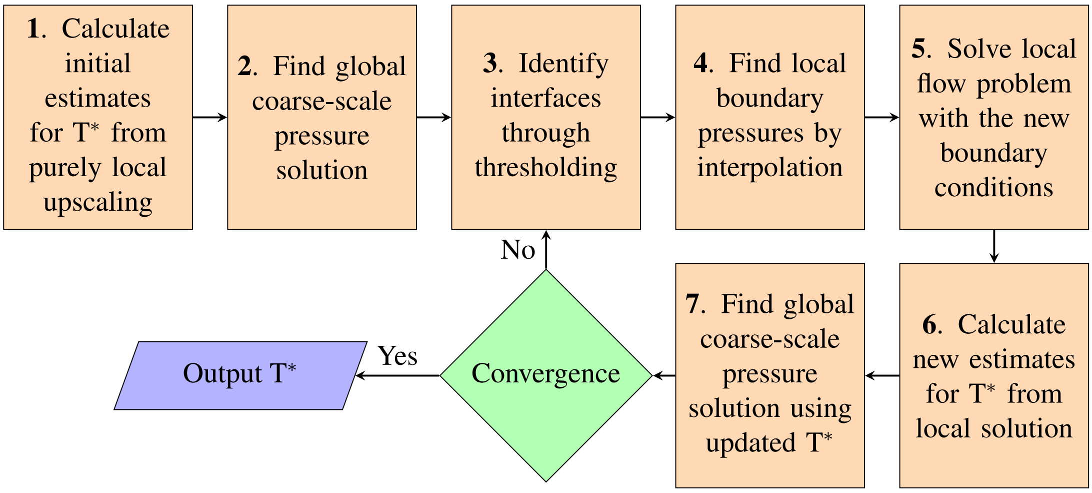

1. Calculate initial estimates for T* from purely local upscaling
2. Find global coarse-scale pressure solution
3. Identify interfaces through thresholding
4. Find local boundary pressures by interpolation
5. Solve local flow problem with the new boundary conditions
6. Calculate new estimates for T* from local solution
7. Find global coarse-scale pressure solution using updated T*
Convergence
No
Yes
Output T*

Figure 3.2: General outline of the ALG upscaling procedure.

The initial step in the ALG procedure is a purely local upscaling of the transmissibility. A solution domain like the one in Fig. 2.4b with standard local boundary conditions is then used and the transmissibility is calculated via equation (3.3). When the initial estimates for the upscaled transmissibilities are found, a global pressure solution on the coarse scale is found. The difference between the ALG upscaling and regular global upscaling procedures is thus that the global pressure solutions are found on the coarse scale rather than the fine scale. This limits the computational demand of finding the pressure solutions significantly.

Based on the global coarse scale pressure solution, an iteration procedure is initiated where the upscaled transmissibilities are re-calculated. As mentioned in the previous chapter, a problem often encountered when using global flow information to calculate the upscaled transmissibility is the occurrence of negative and anomalous transmissibilities. In order to limit the occurrences of such transmissibilities and reduce the computational cost of the iteration procedure, thresholding is used to determine which interfaces the transmissibility is to be re-calculated for.

Chen and Durlofsky (2006a) found that the negative and anomalous transmissibilities often occur in regions of low flow. The thresholding should, therefore, constrict the ALG algorithm to only re-calculate transmissibilities in areas of high flow. This is done by finding the interfaces where the coarse-scale flux magnitude is greater than 10 % of the maximum flux magnitude of the fluxes across all coarse interfaces. Another possibility is to identify interfaces for which the absolute flow rate is greater than a quantile of the distribution of the magnitude of the coarse-scale fluxes for similarly oriented faces (Wen et al., 2006).

Once the interfaces violating the threshold are found, the coupling of the local solution domain and the global coarse-scale grid is performed. The grids are coupled by incorporating the coarse-scale pressures obtained from the global pressure solution into the boundary conditions used in the re-calculation of the upscaled transmissibilities. When the transmissibilities are re-calculated, an extended-local domain like the one shown in Fig. 2.7 is used since the use of border regions may often improve the results by including global effects.

The coarse-scale pressures are used to define the boundary conditions in the extended-local solution domain through an interpolation procedure. The interpolation procedure used here differs from the trilinear interpolation used by Wen et al. (2006). Instead of the trilinear interpolation, a highly efficient linear scattered interpolation is used. The scattered interpolation is performed by creating two meshes: a coarse mesh containing the faces corresponding to the coarse-scale pressures and a fine mesh containing the fine-scale faces at the edges of the extended-local solution domain. The coarse-scale pressures are assigned to their respective locations in the coarse mesh, and an interpolant is created based on this mesh through a Delaunay triangulation of the data points (Amidror, 2002). The interpolant is then used to interpolate the coarse-pressures onto the fine-scale mesh.

By using the new boundary conditions, a new transmissibility is calculated based on a pressure solution in the extended-local domain. If the transmissibility is negative or anomalous, it is simply discarded. Once the transmissibility has been re-calculated for all the interfaces identified by the thresholding, a new global coarse-scale pressure solution is found and the process of re-calculation with new boundary conditions is repeated until convergence is obtained. Whether or not convergence is obtained is decided by criteria based on the change of pressure and flow rate from one iteration to another, defined as

$$\frac{\|\delta p^c\|_2}{\|(\delta p^c)^1\|_2} < \beta_p \quad \text{or} \quad \delta Q < \beta_Q \tag{3.5}$$

where β_p and β_Q are the tolerances. The tolerances should be chosen with care, as they have a direct influence on the computational cost of the ALG procedure. Here the tolerances used by Chen et al. (2003) are used, i.e. $\beta_p = 0.01$ and $\beta_Q = 0.001$. The change in pressure from one iteration to another is found as

$$||\delta p^c||_2 = \left[\frac{1}{n_x n_y n_z}\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\sum_{k=1}^{n_z}((p^c_{i,j,k})^{\nu+1} - (p^c_{i,j,k})^{\nu})^2\right]^{1/2} \tag{3.6}$$

where (i, j, k) is the coarse block index, n_x, n_y and n_z are the number of coarse blocks in the x,y and z direction and ν is the iteration step. The change of total flow rate is found as

$$\delta Q = \max\left(\frac{|Q_\xi^{\nu+1} - Q_\xi^{\nu}|}{|Q_\xi^0|}, \frac{|q_w^{\nu+1} - q_w^{\nu}|}{|q_w^0|}, w = 1, ..., n_w\right) \tag{3.7}$$

where ξ is the axial directions (x,y,z) and n_w is the number of wells in the model if wells are present. Notice that the change in pressure are normalized by the pressure change in the first iteration in equation (3.5), and that the change in flow rate is normalized by the initial flow rate in equation (3.7).

3.2 Two-phase upscaling

This section describes the standard flow-based steady-state upscaling procedure used and introduces the tracer-based two-phase upscaling. The upscaled two-phase flow parameters are used in conjunction with the upscaled single-phase parameters from the methods described in the previous sections when coarse-scale simulations are performed. The greatest detail will be given on the flow-based upscaling procedure, since the tracer-based procedure only differs from it in some steps. The description of the tracer-based upscaling will highlight and explain these differences. All code required to reproduce the two-phase upscaling results may be found in Appendix C.

3.2.1 Flow-based upscaling of relative permeability

The flow-based upscaling of relative permeability is the reference solution which the tracer-based upscaling will be compared to. The method is a steady-state method with a similar procedure as the steps by Odsæter et al. (2015) presented in the previous chapter. The function may be found in the steady-state module of MRST as *upRelPerm*. The outline of the function is shown in **Fig. 3.3**.

Due to the computational costs of a global two-phase upscaling, a local upscaling procedure is used. The function then loops through all the blocks in the coarse grid and performs the upscaling procedure described below for all the blocks individually. The upscaling is performed under the assumption of steady-state and incompressible flow.

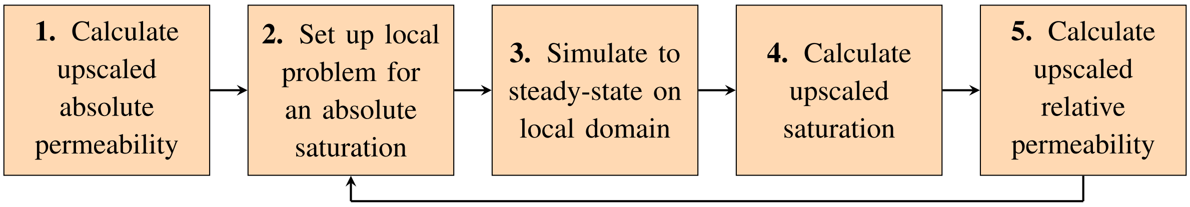

1. Calculate upscaled absolute permeability
2. Set up local problem for an absolute saturation
3. Simulate to steady-state on local domain
4. Calculate upscaled saturation
5. Calculate upscaled relative permeability

Figure 3.3: General outline of the *upRelPerm* function from Lie (2016).

The first step of the flow-based two-phase upscaling is finding the upscaled absolute permeability as shown in Fig. 3.3. This is done in a similar manner as calculating the upscaled transmissibility which was described in the previous section. A single-phase fluid with unit density and viscosity is defined, and the flow is driven by a unit pressure drop along the wanted axial direction as shown in **Fig. 3.4**. Note that for simplicity, Fig. 3.4 shows a two-dimensional domain.

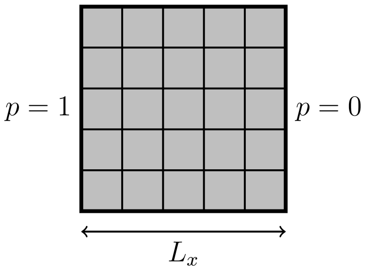


Figure 3.4: Two-dimensional local domain for absolute permeability upscaling.

The pressure equation is solved on the local domain using the pressure solver described in the previous section. Following the pressure solution, the upscaled permeability is found as

$$k_x^* = \frac{u_x L_x}{\Delta p_x} \tag{3.8}$$

for the two-dimensional case shown in Fig. 3.4. This process is repeated for the y- and z-direction if those are wanted as well. Note that when using these boundary conditions, only the diagonal elements of the upscaled permeability tensor are found. If the full tensor is desired, periodic boundary conditions should be used. See e.g. Durlofsky (2005) or Odsæter et al. (2015) for a discussion on periodic boundary conditions for steady-state upscaling.

The next step is to set up the local solution domain corresponding to the fine-scale cells within a coarse grid block. The initial step in this procedure is to populate the fine-scale cells within the grid block with saturations. This is done by defining a vector of absolute saturations, S_w^A. The saturations are denoted as absolute since no consideration is given to the residual saturations of the fine-scale cells at this point. The number of absolute saturations will decide the number of points the upscaling will give on the relative permeability curves. As such, they have a direct influence on the computational cost of the procedure and the number of points should be chosen with care.

Once the fine-scale cells in a grid block has been given an absolute saturation value, the absolute saturation have to be corrected for the residual saturations in the fine-scale cells as

$$S_w = S_{wir} + S_w^A[(1 - S_{or}) - S_{wir}] \tag{3.9}$$

where the S_{wir} is the irreducible water saturation and S_{or} is the residual oil saturation. An example of the residual saturation correction is shown in **Fig. 3.5** where the absolute saturation is shown on the left, while the saturation has been corrected for residuals in the middle. Note that if the absolute saturation is equal to 0 or 1, an endpoint upscaling is performed instead of simulations to steady-state. The upscaled residual saturation is then found as the pore-volume average of the fine-scale residual saturations, while the upscaled endpoint relative permeability of water is found by using the fine-scale relative permeability data at $S_w = 1 - S_{or}$.

In cases where the absolute saturation is not equal to 0 or 1, flow simulations are performed until steady state is reached. In order to perform the simulations, a new set of boundary conditions have to be set. As discussed in the previous chapter, the choice of boundary conditions can have a great influence on the accuracy of the results. Since the objective of the work presented here is a comparison of the results from different upscaling procedures, the comparison of the upscaled results is more important than obtaining highly accurate results with regards to the fine-scale solution. Therefore, a unit pressure drop is chosen for both the flow-based and the tracer-based upscaling considered in the next section.

Since two-phase simulations are performed, saturation boundary conditions also have to be specified. The saturation boundary conditions are set as the saturation in the cells neighboring the outer faces of the grid block. These saturation conditions are kept fixed for the simulations to steady-state such that the resulting saturation distribution is not known in advance of the simulations. This is illustrated in the middle and right figure of Fig. 3.5 where the saturations at the boundaries are shown inside the thick lines. The simulations to steady-state are performed by using a fully implicit auto differentiation solver (Lie, 2016). The solver will solve time steps until convergence is reached, or until the maximum number of iterations is reached. If the maximum number of iterations are reached, the steady-state was not found.

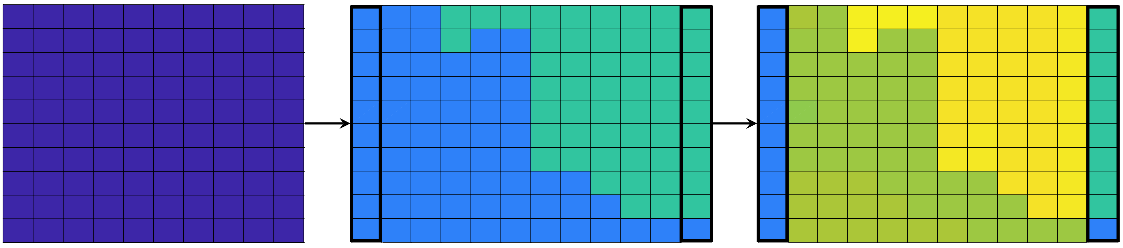


Figure 3.5: Illustration of absolute saturation (left), absolute saturation corrected for residuals (middle), and steady-state saturation distribution (right). The cells covered by a thick border are illustrations of the saturation boundary conditions which are used in the simulations to steady state.

The convergence criteria are based on the change of pressure, flux and saturation from one time step to another with the maximum absolute row sum defined as

$$\|X\| = \max_i \left(\sum_{j=1}^{n} \mid x_{ij}^{\nu+1} - x_{ij}^{\nu} \mid \right) < \epsilon \tag{3.10}$$

where X is a dataset, x is the element in row i and column j of the dataset, ϵ is a tolerance, and ν is the time step. The tolerances ϵ are set to 10, 10^{-4} and 10^{-8} for the pressure, saturation and flux respectively. The simulations will give the steady-state saturation distribution, for which the upscaled saturation is found as

$$S_w^c = \frac{1}{V_p} \sum_{i=1}^{N} V_{pi} S_{wi} \tag{3.11}$$

where V_p is the total pore volume of the coarse grid block and S_{wi} and V_{pi} are the saturations and pore volumes of the fine-scale cells within the block. The steady-state saturation distribution is also used to find the relative permeability distribution of the fine-scale cells. The oil relative permeability is found by using the fact that $S_o = 1 - S_w$. The fine-scale relative permeabilities are then multiplied with the absolute permeability of the fine-scale cells to give the effective phase permeability

$$\mathbf{k}_p = k_{rp}\mathbf{k} \tag{3.12}$$

The next step is finding the upscaled effective phase permeability. In cases where all $k_{rp}\mathbf{k}$ are equal to zero, there will be no fluid motion and the upscaled phase permeability is set to zero. Additionally, to avoid a singular matrix when finding the upscaled phase permeability, zero permeabilities are set to something larger than zero in cases where not all $k_{rp}\mathbf{k}$=0. The upscaled phase permeability is then found by using the same methodology as for the upscaled absolute permeability, with the fine-scale permeability field set to $k_{rp}\mathbf{k}$. The upscaled relative permeability may then be found as

$$k_{rp}^* = \frac{\mathbf{k}_p^*}{\mathbf{k}^*} \tag{3.13}$$

where $\mathbf{k}^*$ is the upscaled absolute permeability. The process above is performed for both oil and water, and for each axial direction wanted. When upscaled relative permeabilities are found for both oil and water for all the absolute saturations, a check is performed to make sure all are inside the valid range. Relative permeabilities outside the range, i.e. $k_{rp} > 1$ or $k_{rp} < 0$, are simply forces inside the valid range.

When the upscaled relative permeabilities with the coherent upscaled saturations are found, the relative permeabilities are interpolated to find the upscaled relative permeability curves. The interpolation method used is a shape-preserving piecewise cubic interpolation which interpolates using a piecewise cubic polynomial $P(x)$.

The polynomial $P(x)$ is a Hermite interpolating polynomial on each sub interval $x_k \leq x \leq x_{k+1}$ (MathWorks, 2019). Hermite interpolation creates an interpolant which is based both on the input values and their derivatives which means that $n(m+1)$ values have to be known as shown below (Burden and Faires, 2010).

$$\begin{array}{llll}
(x_0, y_0), & (x_1, y_1), & \ldots, & (x_{n-1}, y_{n-1}), \\
(x_0, y_0^{'}), & (x_1, y_1^{'}), & \ldots, & (x_{n-1}, y_{n-1}^{'}), \\
(x_0, y_0^{''}), & (x_1, y_1^{''}), & \ldots, & (x_{n-1}, y_{n-1}^{''})
\end{array}$$

The resulting interpolant is shape preserving. This means that the slopes at a point are chosen such that the polynomial preserves the shape of the input data and respects monotonicity. The polynomial is therefore monotonic where the data points are, and when the data has a local extremum so does the polynomial (MathWorks, 2019). The shape-preserving feature of the interpolation is ideal for relative permeability interpolation since the shape of the relative permeability curves will influence the fluid motion. The code for the interpolation procedure may be found in Appendix C as **Code C.2**.

Some additional post-processing of the upscaled relative permeability is required before they can be used in coarse-scale simulations. Since the upscaling procedure is a local one, the result will be one relative permeability curve for both oil and water for each grid block. These curves thus have to be assigned to the correct grid block. This is done by defining N coarse-scale regions, where each region corresponds to a coarse grid block as shown in **Code C.4**.

3.2.2 Tracer-based upscaling of relative permeability

The tracer-based upscaling is a new alternative to the flow-based upscaling described in the previous section. The basic idea of the method is to predict the steady-state saturation distributions based on tracer simulations. The outline of the method is shown in **Fig. 3.6**. The procedure is similar to the procedure shown in Fig. 3.3 and a lot of the methodology will be explained with reference to the one presented in the previous section. In fact, the only differences between the methods occur in steps 3 and 4.

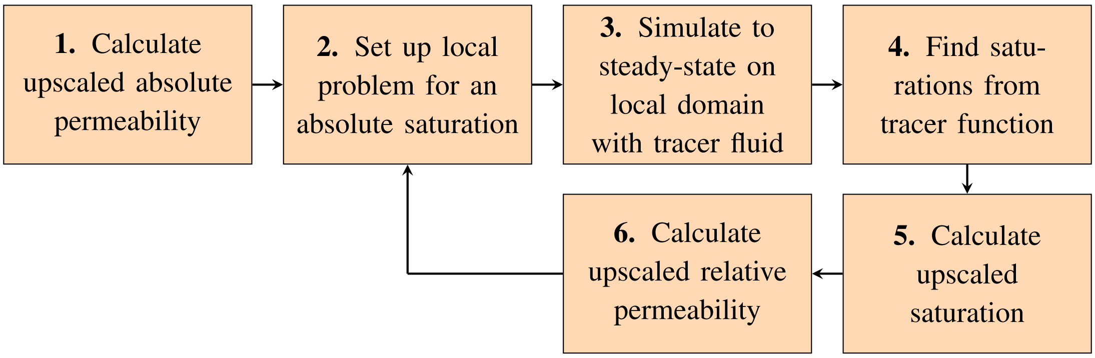

1. Calculate upscaled absolute permeability
2. Set up local problem for an absolute saturation
3. Simulate to steady-state on local domain with tracer fluid
4. Find saturations from tracer function
5. Calculate upscaled saturation
6. Calculate upscaled relative permeability

Figure 3.6: General outline of the tracer-based two-phase upscaling.

As discussed in the previous chapter, finding the saturation distribution is the most demanding part of steady-state flow-based upscaling, since it requires full flow simulations to reach steady-state. Simplifications can be made if the flow is very low or very high with the capillary- or viscous-limit methods, but the conditions required for these methods to be feasible are not always present in the entire reservoir. The tracer-based upscaling serves as an alternative to these simplifications, with the major difference being that it is applicable for all conditions.

The first two steps of the tracer-based upscaling procedure are almost equal to those in the flow-based upscaling, with the only difference being that tracer saturations are populated instead of actual saturations. As mentioned in the section introduction, the main difference between the two methods occurs in step 3 when the simulations to steady-state are performed. In the tracer-based method, the original fluids (oil and water) are replaced by a tracer-fluid. This is achieved by letting $\rho_w = \rho_o = 1$, $\mu_w = \mu_o = 1$, $k_{rw} = S_w$ and $k_{ro} = 1 - S_w$. As described in the previous chapter, this amounts to the injection of a tracer. By doing so, the computational costs of the simulations are greatly reduced. In cases where the absolute tracer saturation is equal to 0 or 1, no simulations are performed and an endpoint upscaling is performed as described in the previous section.

The simulations will give the steady-state tracer saturation distribution. A pre-defined function $f(S_t)$ is then used to convert the tracer saturation distribution to the actual water saturation distribution. The function used in this work will give the normalized water saturation with respect to the residual saturations, which means that they have to be un-normalized by using equation (3.9). An example of the conversion from tracer saturations to actual saturations is shown in **Fig. 3.7**. The function has to be chosen such that the output values are as close to the actual saturation distribution as possible. The function's impact on the results will be discussed in a later chapter.

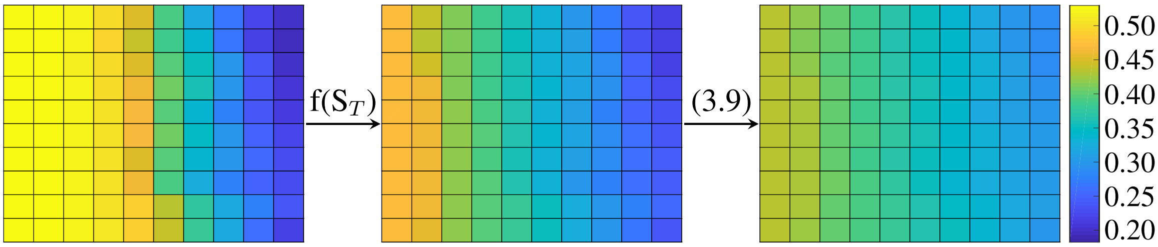


Figure 3.7: Illustration showing the steady-state tracer saturation distribution on the left, the saturation distribution obtained from the tracer function in the middle and the un-normalized saturation distribution on the right.

The tracer functions, $f(S_T)$, are based on an interpolation of data from regular and tracer flow simulations. When interpolation is performed in MATLAB the data have to be unique, which is not always the case for saturation data since multiple cells may have equal tracer and water saturation. The data used to create the functions are data samples from multiple time steps such that the datasets grow very large and multiple occurrences of similar or equal saturations are frequent. In order to ensure the uniqueness of the data, the tracer saturation distributions are partitioned into 100 bins with a uniform width as shown in **Fig. 3.8**. When the data is binned, the count of data points within each bin is recorded.

The bins are created by making sure that the sum of the values within the bins are equal to or less than the number of input tracer saturations. The binning procedure will make sure that the bins cover the range of tracer saturations and reveal the underlying shape of the distribution. The bins themselves are made up of equally spaced intervals of tracer saturations such that when an average saturation value in each bin is found it is assigned to the tracer saturation that is the average of the edges of each bin. Consider the largest bar in Fig. 3.8 as an example: the edges of the bin are 0 and 0.01 and the tracer saturation for which the average water saturation within the bin is found for is thus 0.005. The average saturation value within each bin is found by summing all the water saturations that fall within the bin and dividing by the total number of counts in the bin.

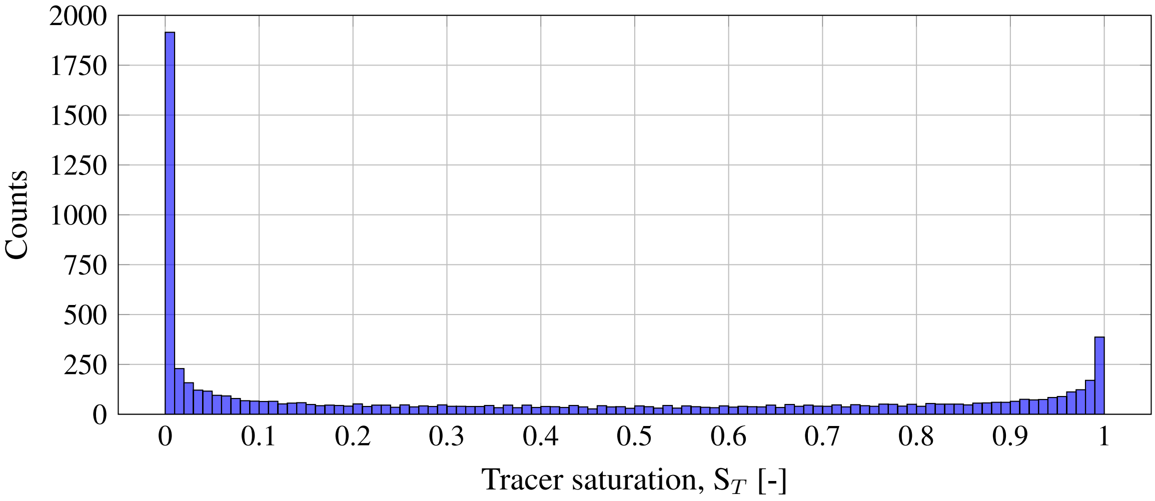


Figure 3.8: Illustration of the binning in the tracer function generation procedure.

Since the bins are chosen such that the shape of the distribution of the dataset is preserved, the averaging described above will give an averaged dataset with a distribution equal to that of the original dataset as shown in **Fig. 3.9**. This is important since the saturation distribution will influence the upscaled relative permeability curves, as opposed to every cell within a block having the same saturation. Once the averaged dataset has been found, the function is created by a linear interpolation of the averaged data.

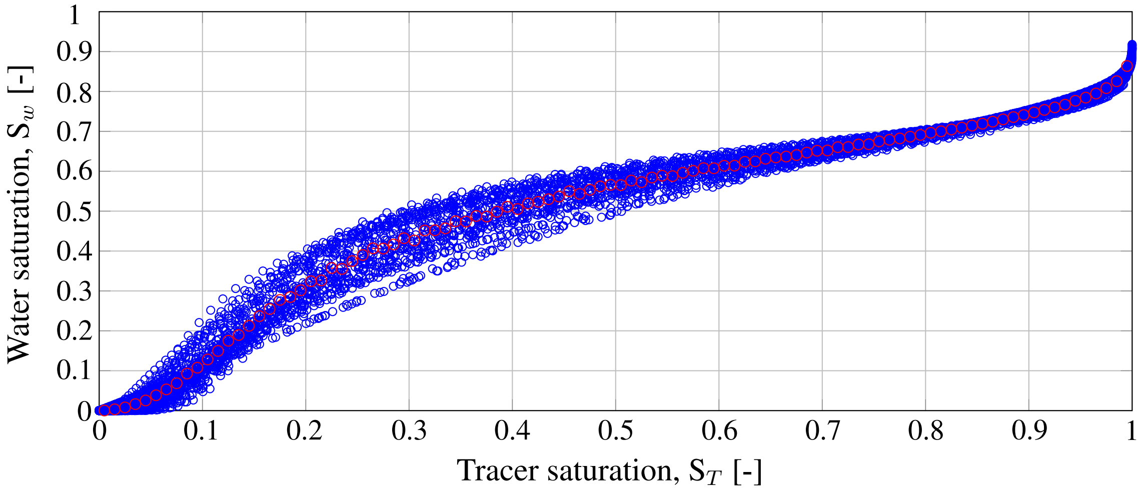


Figure 3.9: Raw data used to generate the tracer functions shown in blue, and the resulting points from the averaging procedure for which the function is created shown in red.

The calculation of the upscaled relative permeability in the tracer-based method is equal to the procedure in the flow-based method, i.e. the upscaled relative permeability is calculated by using the saturation distribution obtained from the tracer function in conjunction with the fine-scale relative permeability curves. The only difference between the methods is thus how the steady-state saturation distribution is found.

As will be demonstrated in a later chapter, the use of tracer-based upscaling limits the computational cost while maintaining accuracy in the upscaled results compared to the flow-based method. This is shown through simulations on different reservoir models with different flow regimes. The code required for the tracer-based upscaling may be found in Appendix C.

3.2.3 Upscaling of capillary pressure

In addition to finding the upscaled relative permeability curves, an upscaling of the capillary pressure curves is performed in conjunction with both methods described above in the *TwoPhaseUpscaler* class from MRST which is used to perform the upscaling. The upscaling is performed by using the *upPcOW* function in MRST who's outline is shown in **Fig. 3.10**. It should be noted that the capillary pressure is upscaled by the capillary limit method which was described in the previous chapter, simply because it is the default setting of the upscaler class. The effect of including and upscaling capillary pressure will be discussed in Chapter 5.

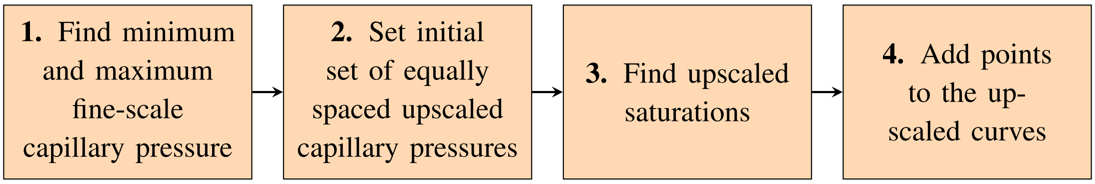

1. Find minimum and maximum fine-scale capillary pressure
2. Set initial set of equally spaced upscaled capillary pressures
3. Find upscaled saturations
4. Add points to the upscaled curves

Figure 3.10: General outline of the *upPcOW* function from Lie (2016).

The first step in the capillary pressure upscaling procedure is finding the minimum and maximum capillary pressure in the cells within a block. This is done by finding the minimum and maximum value when all cells have a water saturation of 0 and 1 under the assumption of negligible gravity. Based on the minimum and maximum value, an initial set of equally spaced upscaled capillary pressures is defined such that

$$p_{c,i}^{c} = [\min(p_c(S_w = 0), p_c(S_w = 1)), ..., \max(p_c(S_w = 0), p_c(S_w = 1))] \tag{3.14}$$

The next step is finding the upscaled saturation which correspond to the initial upscaled capillary pressures. This done is by giving all the cells within a block the same capillary pressure value from the initial upscaled values. The fine-scale saturation distribution is then found by inverting the fine-scale capillary pressure curves and using the value from the initial upscaled set as the input value. The upscaled saturation is then simply found as the pore volume-average of the fine-scale saturation distribution. This process is repeated for all values in the initial set of equally spaced upscaled capillary pressures. The result will then be an initial set of saturation and capillary pressure pairs.

In order to make sure that there are no big jumps between the initial pairs of saturation and capillary pressure, a relative tolerance of 0.01 is defined as the maximum difference between points. A check is then performed to make sure that the maximum relative difference between any two pairs is less than the tolerances. The relative difference is defined for both saturation and capillary pressure values as

$$\|d_S\| = \frac{d_S}{S^c_{w,\max} - S^c_{w,\min}}, \quad \|d_{p_c}\| = \frac{d_{p_c}}{p^c_{c,\max} - p^c_{c,\min}} \tag{3.15}$$

where d denotes the difference between two pairs. If the maximum relative difference of either saturation of capillary pressure is larger than the relative tolerance, a new pair of upscaled saturation and capillary pressure is found in between the pairs with the maximum relative difference.

The procedure of finding the new pair is similar to that used for the initial pairs. The only difference is that the cells within the block are given a capillary pressure value equal to the mean of the values in the pairs which gave the maximum relative difference. The initial set of pairs is refined until none of the relative differences are greater than the tolerances, or if a maximum number of iterations are met.

The effect of minimizing the differences between the pairs is shown in **Fig. 3.11**, which shows that the curve with the refined points has a smoother shape than the initial set. For the case shown in Fig. 3.11 the differences between the curves are not significant such that a refinement may not the necessary. In cases where the initial set has large jumps between the points, however, a refinement is important to obtain accurate upscaled capillary pressure curves.

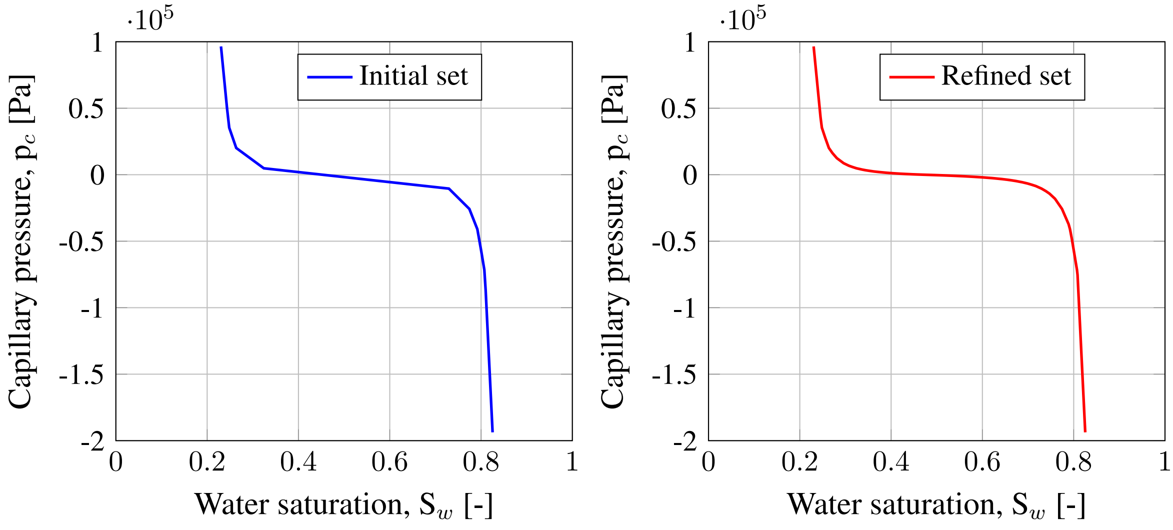


Figure 3.11: Illustration of the difference between the initial and refined upscaled capillary pressure curves.

The refinement process may give pairs of equal upscaled water saturation. In order for the upscaled capillary pressure curves to be monotonic and to avoid jumps in the curves, the duplicate saturations are removed from the refined set. A check is also made to make sure that the upscaled saturations are increasing. If they are not, the refined set of saturations and capillary pressures are simply flipped or sorted to ensure monotonicity.

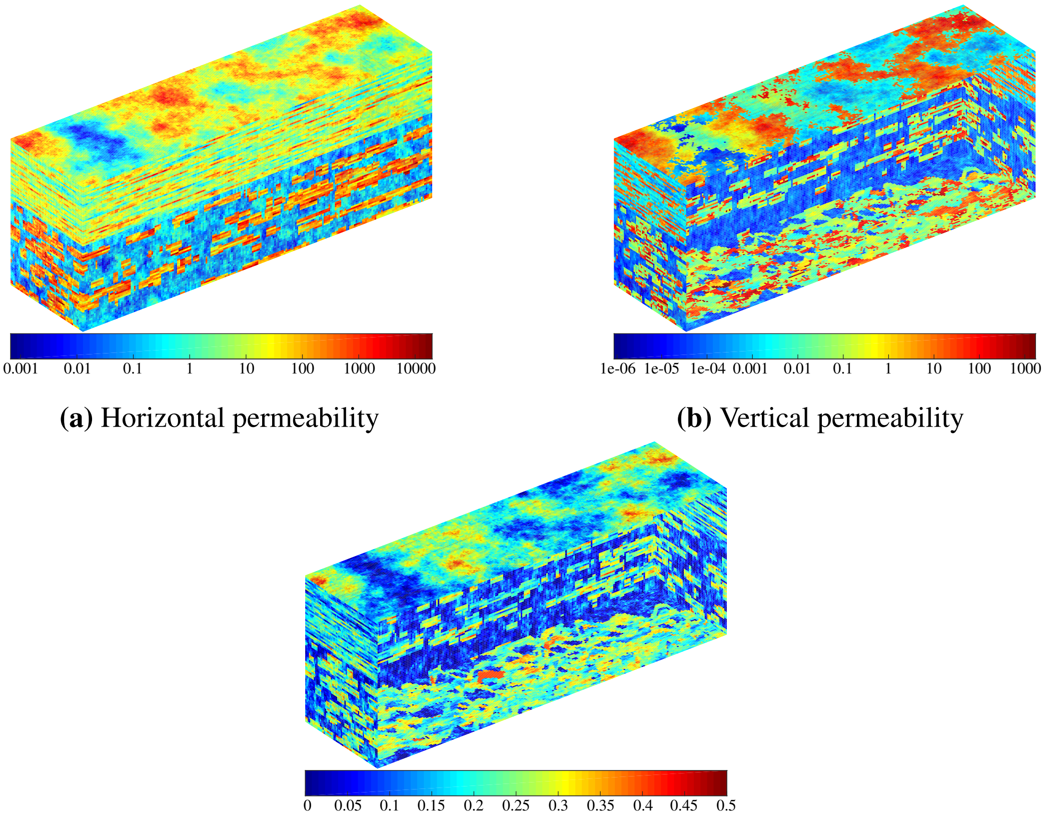
Note that the process described above is performed under the assumption that the fine-scale capillary pressure curves are monotone.

3.3 Simulation model

The simulations are performed on the SPE10 model introduced by Christie and Blunt (2001). The model consists of two formations: layer 1-35 represents the Tarbert formation, while layer 36-85 represents the Upper Ness formation. The Tarbert formation is a prograding near-shore environment consisting of sandstone, siltstone, and shale. The Upper Ness formation is a fluvial formation deposited in a delta-plain continental environment with high permeable sandstone channels surrounded by a low permeable background of shale and coal.

Since there is a background of nearly impermeable shale and coal in the Upper Ness formation, there is little communication between the different sandstone channels. These channels are ideal for testing the accuracy of two-phase upscaling, since only a small portion of a coarse grid block may be subjected to flow. In such cases, single-phase upscaling alone is often not sufficient to characterize fluid displacement since the large scale effects of the relative permeability have to be included in the upscaled model (Aarnes et al., 2007).

As seen in **Fig. 3.12c** there is a large span in the porosity values in the model, and there are cells with zero porosity which should be considered inactive. The model has a simple geometry, which makes it ideal for comparing different upgridding and upscaling techniques. The permeability of the model shown in Figs. 3.12a and 3.12b is a diagonal tensor with equal permeabilities in the horizontal directions.



(a) Horizontal permeability

(b) Vertical permeability

(c) Porosity

Figure 3.12: Permeability and porosity of the SPE10 model.

To make the upscaling more interesting, the model is divided into two regions with different relative permeability data. Cells, where $\log(k)$ is greater than the mean of the logarithm of the permeability, are assigned to region 1, while the rest of the cells are assigned to region 2. **Fig. 3.13** shows the regions, with region 1 in blue and region 2 in yellow.

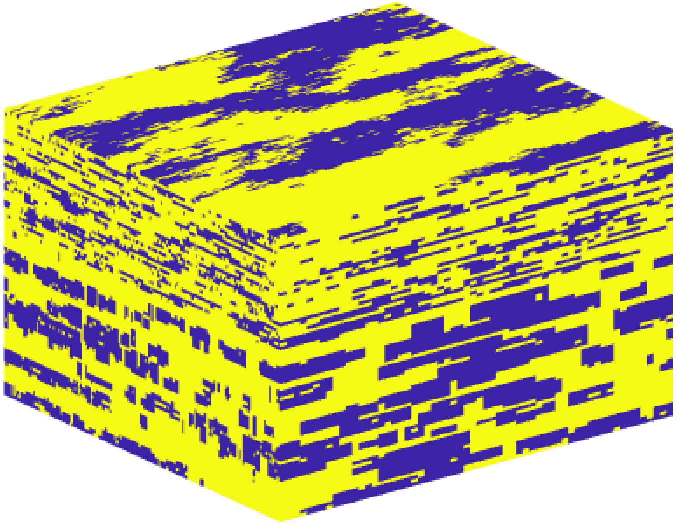


Figure 3.13: Visualization of regions used in simulation model.

Fig. 3.14 shows the relative permeability curves for the two regions which are modeled as Corey curves with $n_w = n_o = 2$, while **Table. 3.1** shows the residual saturation and endpoint relative permeability data for the regions.

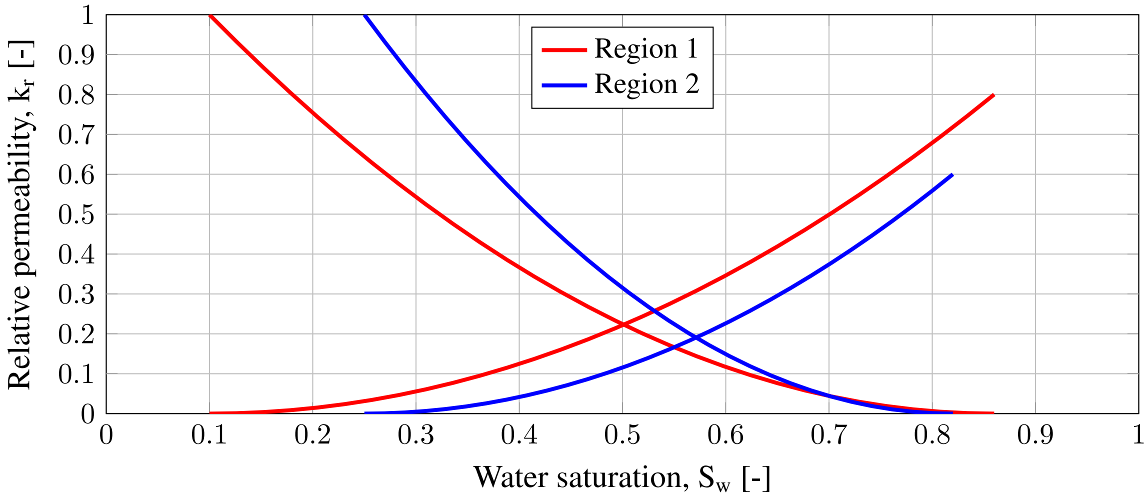


Figure 3.14: Relative permeability curves for the two regions used.

Parameter	**Symbol**	**Region 1**	**Region 2**
Irreducible water saturation	S_{wir}	0.10	0.25
Residual saturation of oil	S_{or}	0.14	0.18
Endpoint relative permeability of water	$k_{\text{rw, max}}$	0.80	0.60

Table 3.1: Residual saturation and endpoint relative permeability data for the regions.

The capillary pressure curves of the fine-scale model are set by using equation (2.50), where the Leverett J-function is modelled as

$$J_L = \frac{1}{S_w^2}, \quad J_R = \frac{1}{(1-S_w)^2}, \quad J = 0.9\frac{J_L}{\max(J_L)} - 1.8\frac{J_R}{\max(J_R)} \tag{3.16}$$

The capillary pressure curves of the fine-scale model are shown in **Fig. 3.15**. Note that capillary pressure is added to the model as a part of the region generation in the *getExampleFluidProps* function in MRST. Since capillary pressure curves are not present in the original setup of the model, simulations without capillary pressure curves will also be performed in order to compare the results.

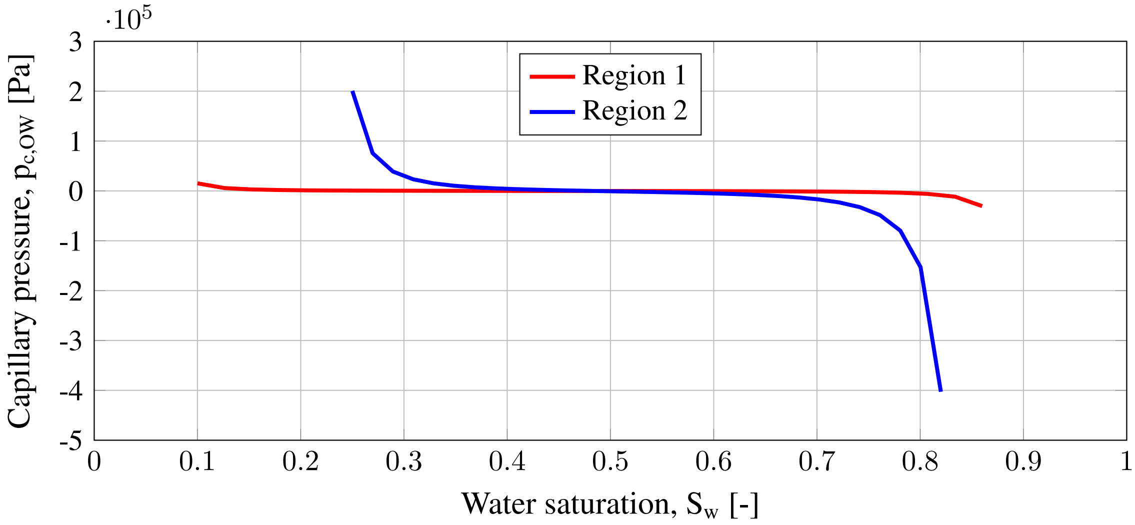


Figure 3.15: Capillary pressure curves for the two regions used.

Flow is driven by a linear pressure difference in the x-direction. The system is initially saturated with oil, i.e. $S_w = S_{wir}$, and water is injected with a constant injection rate from the left edge. The grid is coarsened by a degree of 10 when creating the coarse grid. The fluid properties used for both the fine and coarse scale models are displayed in **Table. 3.2**.

Parameter	**Symbol**	**Value**
Density, water	ρ_w	1014 kg/m^3
Density, oil	ρ_o	850 kg/m^3
Viscosity, water	μ_w	1 cP = 0.001 Pa s
Viscosity, oil	μ_o	1.5 cP = 0.0015 Pa s
Interfacial tension	σ	22 dyne/cm = 0.022 N/m
Contact angle	θ	45°

Table 3.2: Fluid data used for all models.

CHAPTER 4

RESULTS

The results obtained from performing simulations on the upscaled models derived by the procedures described in the previous chapter are presented in this chapter. Results for both single layers and the full model are presented. The upscaled results are compared to the results of the fine-scale model in all plots. A discussion of the results may be found in Chapter 5.

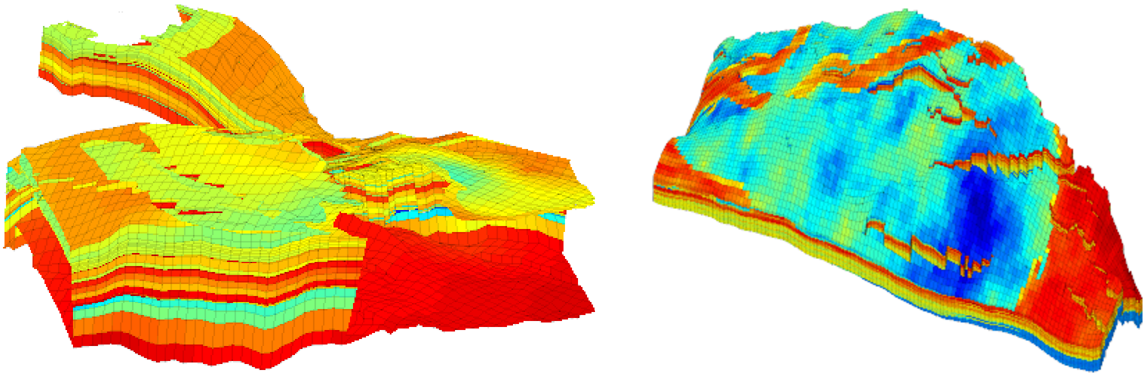
The following nomenclature will be used in the legends when the results are compared in plots:

- **Fine**: Results obtained from fine-scale simulations.
- **Flow-based (*M*)**: Upscaled results obtained from the upscaling procedure described in Section 3.2.1, where M indicates which single-phase upscaling procedure was used. $M =$ G denotes the global transmissibility upscaling while $M =$ ALG denotes the adaptive local-global upscaling procedure.
- **Tracer-based (*M*), f$_N$**: Upscaled results obtained from the upscaling procedure described in Section 3.2.2, where f$_N$ indicates which layer(s) N the tracer function f is derived from. M is defined as for the flow-based upscaling.

Since incompressible flow is assumed, the boundary fluxes must sum to zero if the model is to be well posed. If not, more fluids would be injected than extracted, or vice versa, and hence implicitly violate the assumption of incompressibility (Lie, 2016). Plots of total production rates are therefore not shown since the total production is simply be constant. The results are presented in terms of the water fraction ($F_w = Q_w/(Q_w + Q_o)$) at the outlet versus the number of pore volumes injected (PVI), defined as $\frac{1}{V_p}\int_0^t Q(\tau)d\tau$ where V_p is the total pore volume. In addition to plots, tables with the CPU-time for both simulations and the upscaling procedures will be displayed. A comparison of CPU-times is valuable since one wants the highest possible accuracy at the lowest possible computational cost. Thus, by using the tables in conjunction with the plots a choice on which upscaling procedure that is the most desirable to use may be made.

The chapter is concluded by presenting results obtained by using different tracer functions. In addition to using tracer functions generated based on data from the SPE10 model, functions originating from two other models shown in **Fig. 4.1** are used as well. The additional models are the Norne model which is a sandstone oil and gas field on the Norwegian continental shelf

(NCS), and a model from the Sensitivity Analysis of the Impact of Geological Uncertainties on Production forecasting(SAIGUP) project (Norsk Regnesentral, 2003).



(**a**) Norne

(**b**) SAIGUP

Figure 4.1: Permeability visualization of the additional models used in tracer function generation on the same permeability scale as in Fig. 3.12a.

4.1 Single layers

The initial testing of the tracer-based upscaling procedure was performed on single layers. Results from two single layers are therefore included such that the accuracy of the upscaling for these layers and the full model may be compared. The single layers were chosen based on **Fig. 4.2** which shows the tracer function for every layer in the model.

It is of interest to investigate the sensitivity of the upscaled results to the tracer function. The single layers are therefore chosen to be layer 26 and 44, which corresponds to the layers with the tracer functions on the edges of the stack shown in Fig. 4.2. The sensitivity with respect to the tracer function may then be investigated by upscaling layer 26 with the tracer function from layer 44 and vice versa.

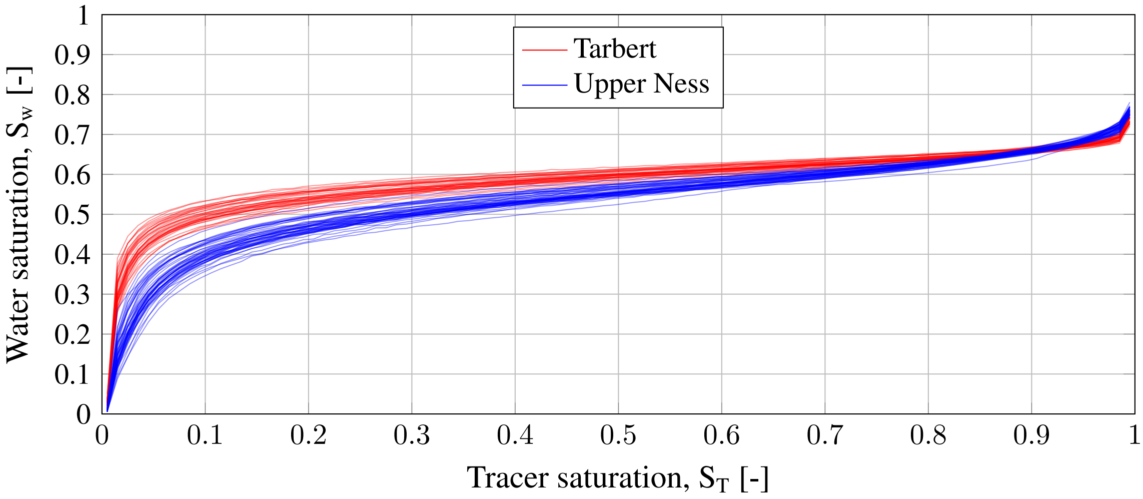


Figure 4.2: Tracer functions for all layers in the SPE10 model.

Fig. 4.3 and **Fig. 4.4** shows a comparison of the water cut at the outflow boundary for layer 26 and layer 44 respectively.

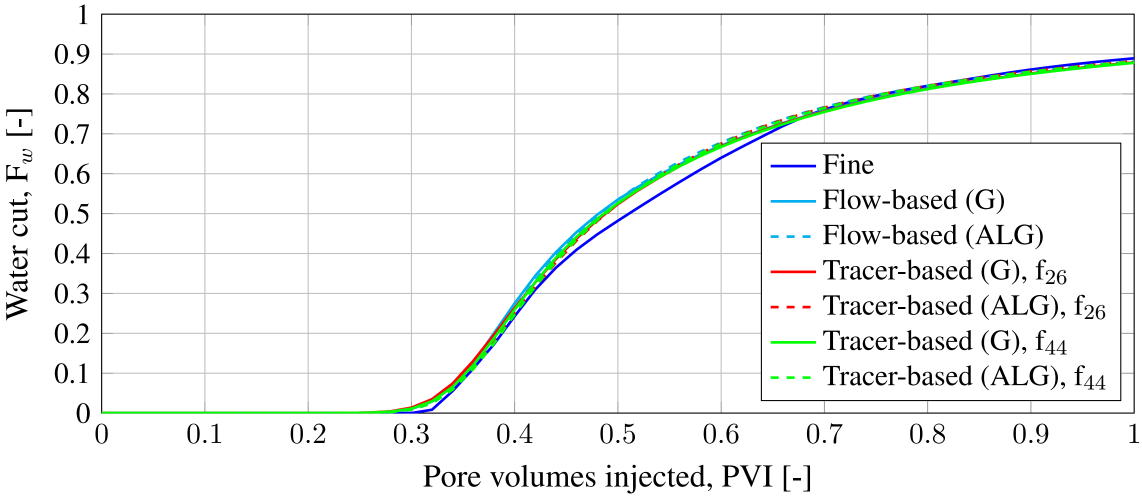


Figure 4.3: Comparison of water cut for layer 26.

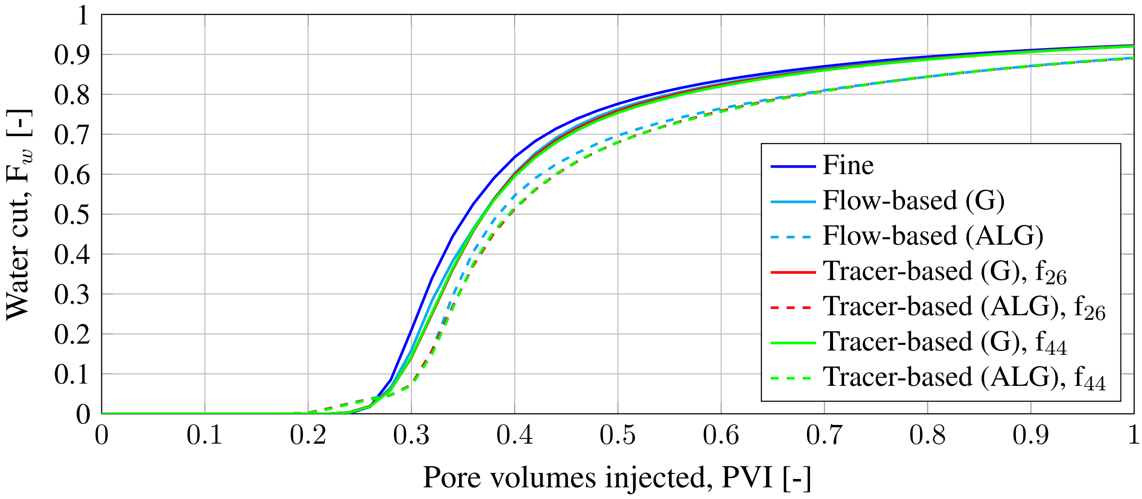


Figure 4.4: Comparison of water cut for layer 44.

The CPU-times for the different operations necessary to obtain the results for the layers are shown in **Table. 4.1**.

Operation	**Layer 26**	**Layer 44**
Fine-scale simulation	1	1
Coarse-scale simulation, flow-based	0.2420	0.3701
Coarse-scale simulation, tracer-based	0.2421	0.3173
Flow-based upscaling (G)	1	1
Tracer-based upscaling (G)	0.3031	0.2792
Flow-based upscaling (ALG)	1.0107	1.0038
Tracer-based upscaling (ALG)	0.3167	0.2829

Table 4.1: CPU-times for the different operations for both layers.

In Table. 4.1, the simulation times are normalized with respect to the fine-scale simulation and the upscaling procedures are normalized with respect the flow-based upscaling where the global single-phase upscaling procedure is used.

A comparison between the fine-scale relative permeability curves and the flow-based upscaled relative permeability curves are shown in **Figs. 4.5** and **4.6**. The relative permeability curves from the tracer-based upscaling are similar to those obtained from the flow-based upscaling and are therefore included in Appendix B.

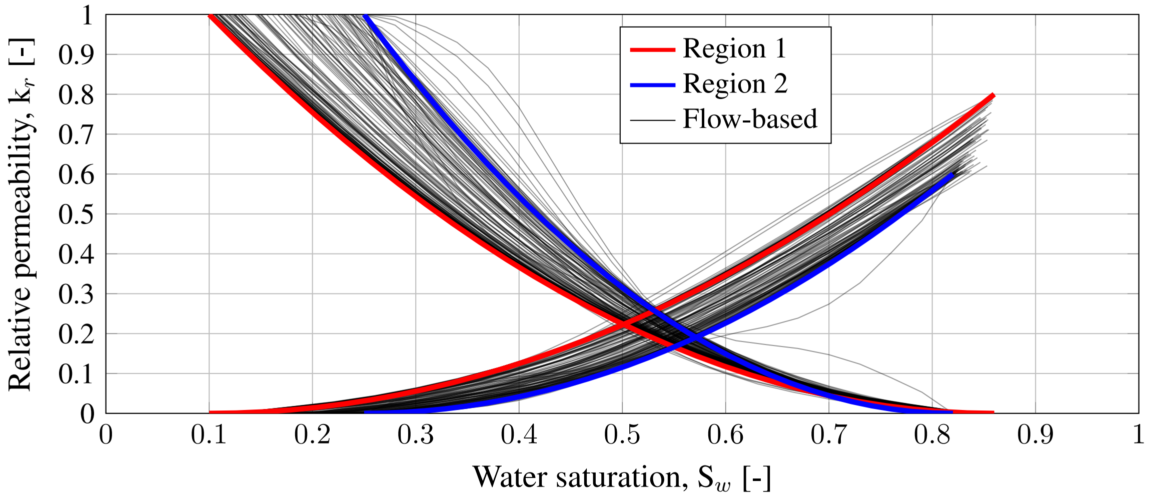


Figure 4.5: Comparison of upscaled and fine-scale relative permeability curves for layer 26.

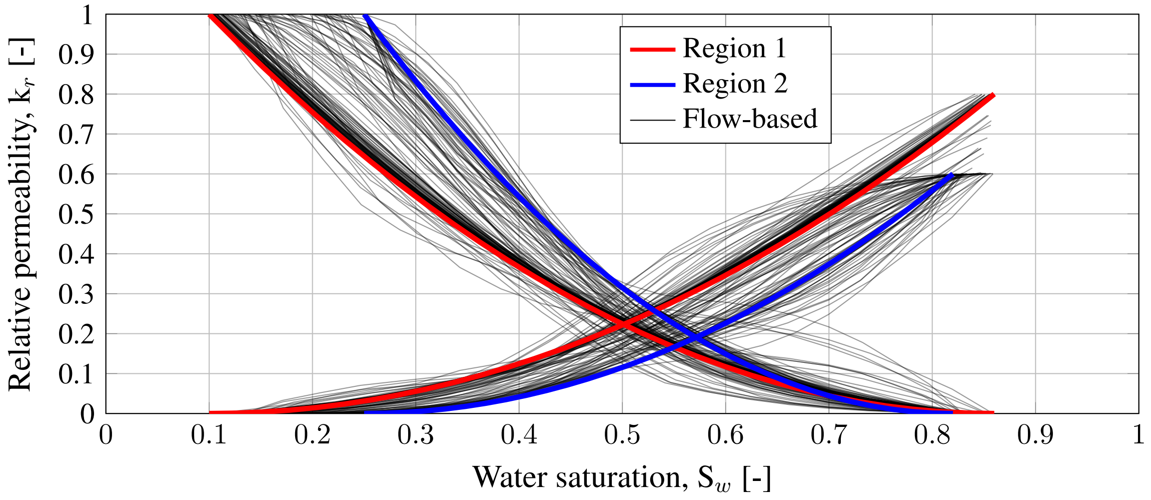


Figure 4.6: Comparison of upscaled and fine-scale relative permeability curves for layer 44.

Fig. A.2 and **Fig. A.3** in Appendix A show snapshots of the saturation distributions at different PVI. In all visualizations in Appendix A, the fine-scale solution is averaged over each coarse block and plotted on the coarse grid. The left column in Figs. A.2 and A.3 is the fine-scale results, the middle column is the upscaled tracer-based results obtained by using the function corresponding to the layer in question and the right column is the upscaled flow-based results.

4.2 Full model

The results for the full SPE10 model are displayed in this section. These results make an interesting comparison with the results for the single layers since the error will tend to accumulate, especially the error resulting from the tracer function when using the tracer-based upscaling. The accuracy of the upscaled models should, therefore, be expected to be lower than for the single layers.

Similarly as for the single layers, **Fig. 4.7** and **Fig. 4.8** shows the water cut results for the full model. The tracer-based results were obtained by using a tracer function corresponding to the full model. The function was created by running tracer and regular simulations on the full model and using the procedure described in Section 3.2.2. The water cut results are divided into two plots to limit the number of lines in each plot, such that Fig. 4.7 shows the results when the global single-phase upscaling is used and Fig. 4.8 shows the results when the ALG single-phase upscaling is used.

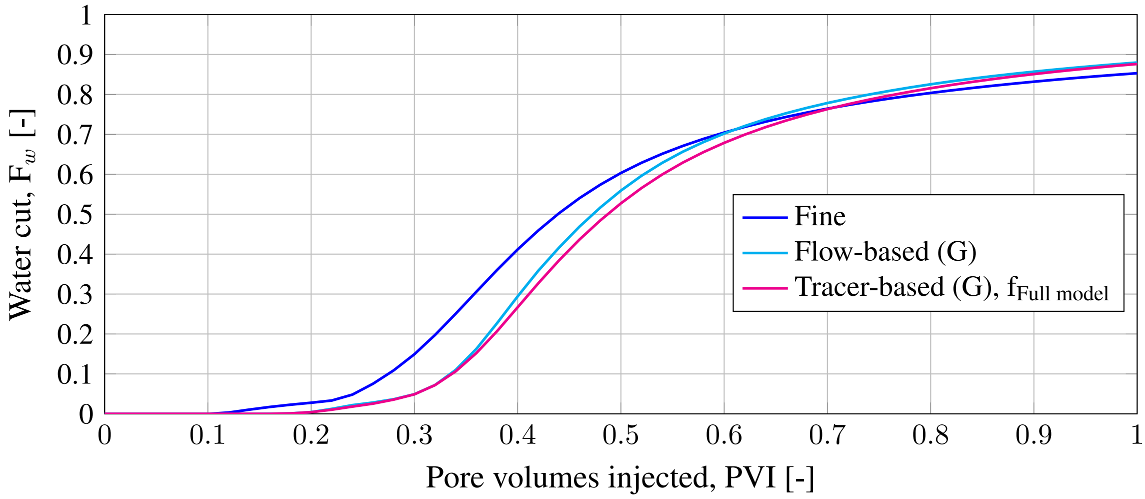


Figure 4.7: Comparison of water cut when global single-phase upscaling is used for the full model.

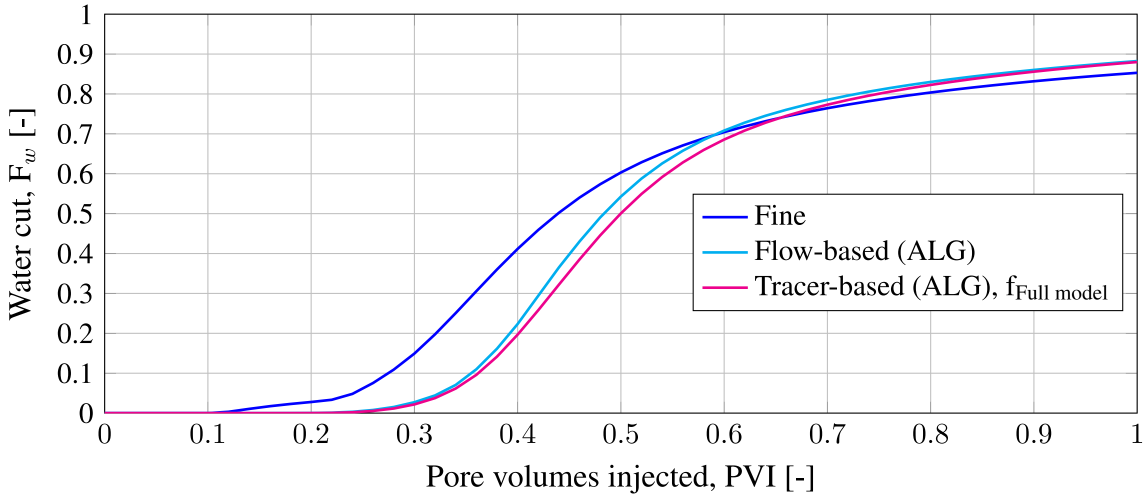


Figure 4.8: Comparison of water cut when ALG single-phase upscaling is used for the full model.

The normalized CPU-times for the different operations in the full model are shown in **Table. 4.1**. The CPU-times are normalized in a similar manner as for the single layers.

Operation	**CPU**
Fine-scale simulation	1
Coarse-scale simulation, flow-based	0.0051
Coarse-scale simulation, tracer-based	0.0062
Flow-based upscaling (G)	1
Tracer-based upscaling (G)	0.3289
Flow-based upscaling (ALG)	1.0325
Tracer-based upscaling (ALG)	0.3614

Table 4.2: CPU-times for the different operations for the full model.

The upscaled relative permeability curves are shown in **Figs. 4.9** and **4.10**.

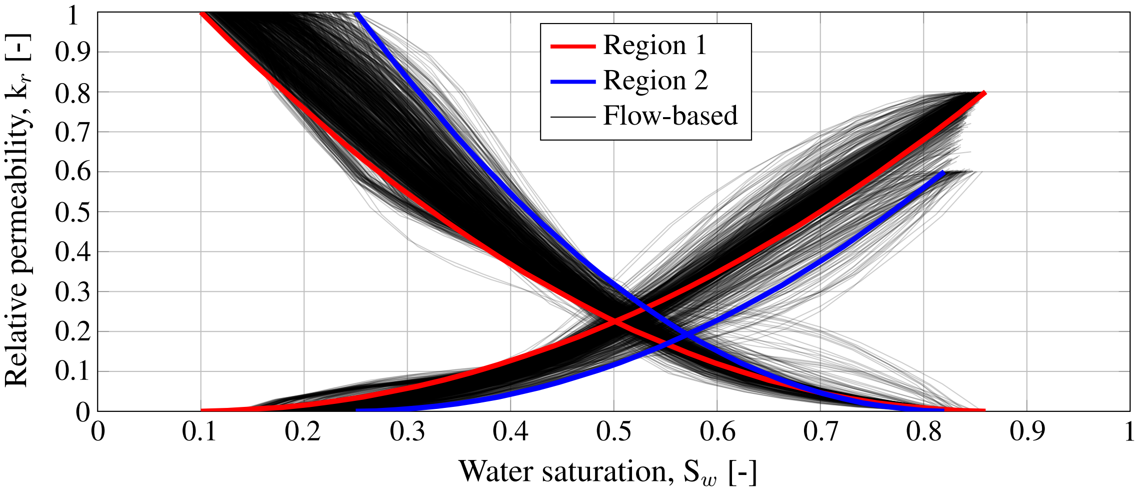


Figure 4.9: Comparison of upscaled and fine-scale relative permeability curves for flow-based upscaling in the full model.

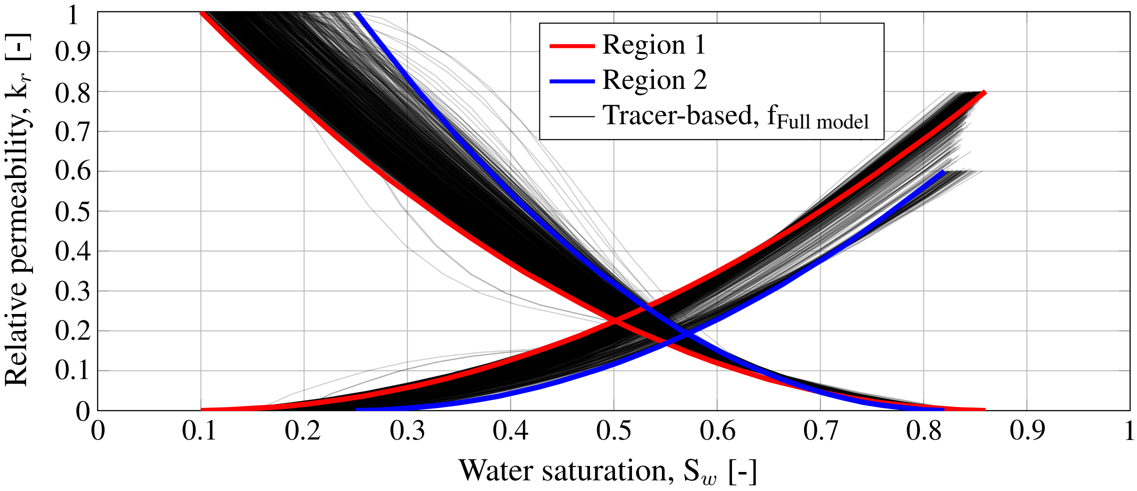


Figure 4.10: Comparison of upscaled and fine-scale relative permeability curves for tracer-based upscaling in the full model.

As previously mentioned, simulations when capillary pressure is neglected for both the fine and coarse models were also performed. **Fig. 4.11** shows the results from these simulations.

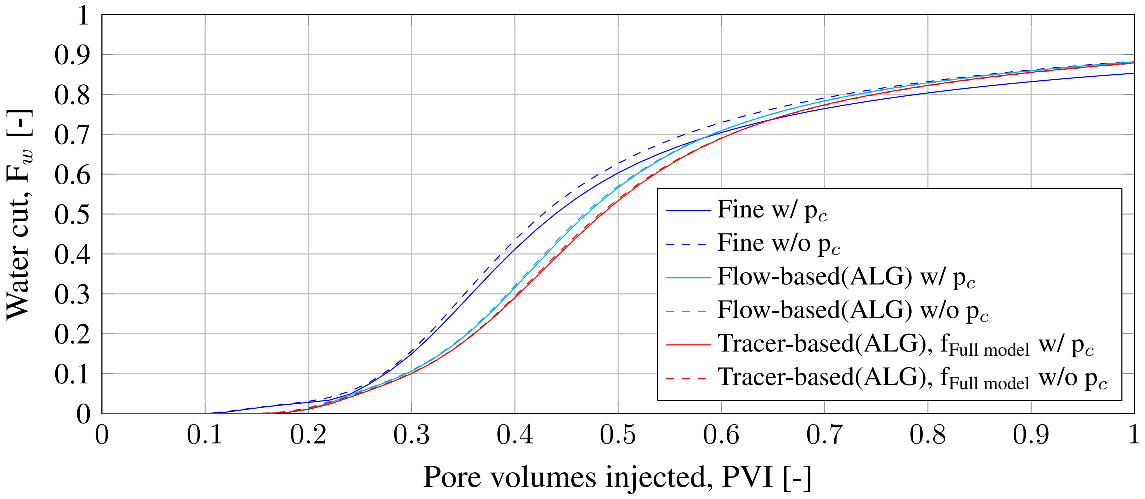


Figure 4.11: Water cut results when capillary pressure is neglected for the full model.

The snapshots of the saturation distributions at different PVI for the full model are shown in **Fig. A.4**. In addition to the saturation distributions, Appendix A also includes plots or the error between the saturation distributions for both the single layers and the full model in **Fig. A.5**. The error in the saturation distributions between the upscaled models are shown in **Fig. A.6**.

In addition to the saturation visualizations, the total coarse-scale flow through the full model is visualized in **Fig. 4.12** and **Fig. 4.13**. Fig. 4.12 shows the total flow when global single-phase upscaling is used, while Fig. 4.13 shows the flow when ALG single-phase upscaling is used. Since incompressible flow is considered, the flow shown in the figures below is constant throughout the simulation as mentioned in the chapter introduction.

The flow visualizations have the same setup as the saturation visualizations in Appendix A: the averaged fine-scale solution on the left, results from a tracer-based upscaling in the middle and results from a flow-based upscaling on the right.

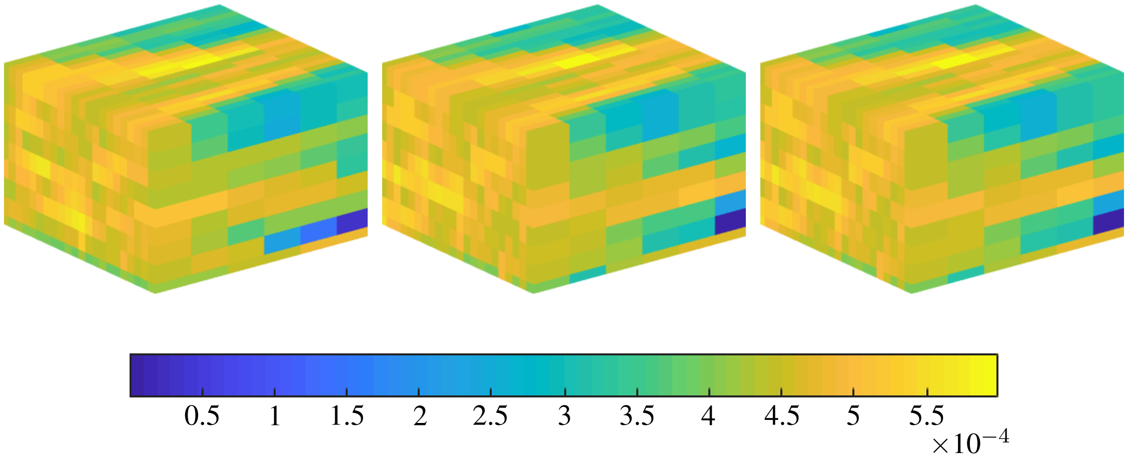


Figure 4.12: Total flow in units [m^3/s] in the full model when global single-phase upscaling is used.

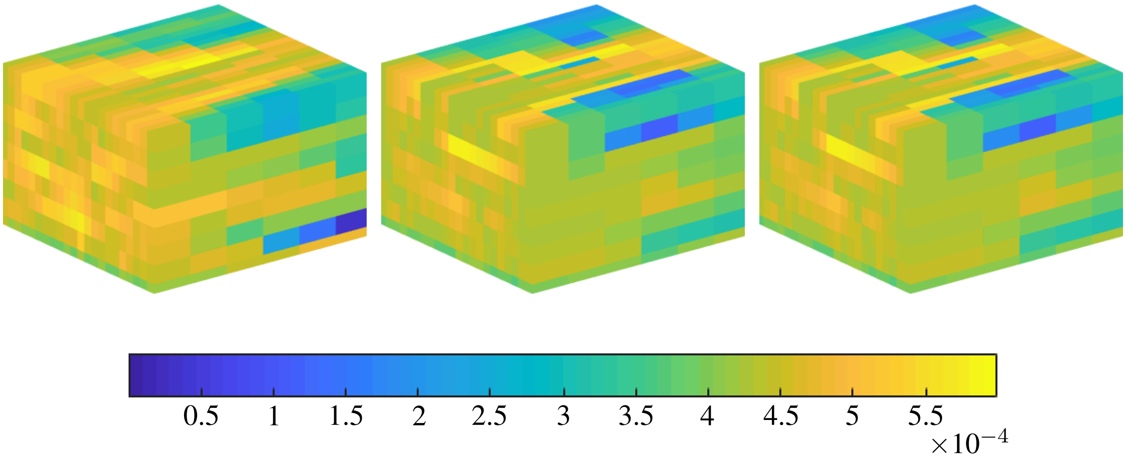


Figure 4.13: Total flow in units [m^3/s] in the full model when ALG single-phase upscaling is used.

4.3 Comparison of results for different tracer functions

As mentioned in the chapter introduction, a sensitivity analysis with respect to the tracer function was performed by comparing results obtained with different tracer functions when performing the tracer-based upscaling procedure in the SPE10 model. **Fig. 4.14** shows all the functions which was used in the analysis.

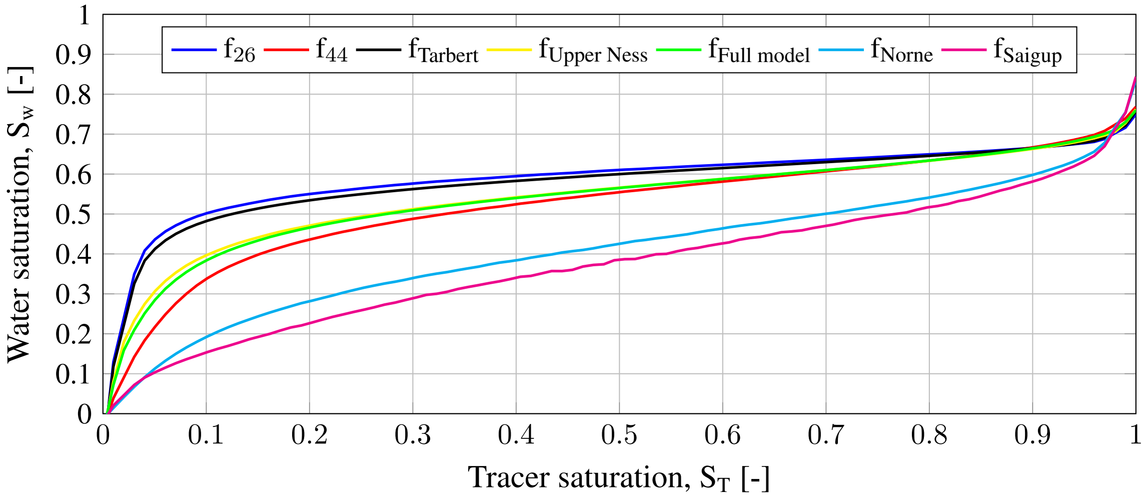


Figure 4.14: All tracer functions used in sensitivity analysis.

The water cut results that was obtained from using the functions in Fig. 4.14 are shown in **Fig. 4.15** and **Fig. 4.16**. Similarly as for the water cut results presented in the previous section, the results obtained by using global and ALG single-phase upscaling are separated to limit the number of lines in each plot. The results that was obtained by using the functions from the Norne and SAIGUP models are not included in the figures simply because they will be close to identical with the other results. The results from these functions may be seen in **Table. 4.3**, where the error with respect to the fine-scale solution from using the different functions in the different cases in the SPE10 model are displayed.

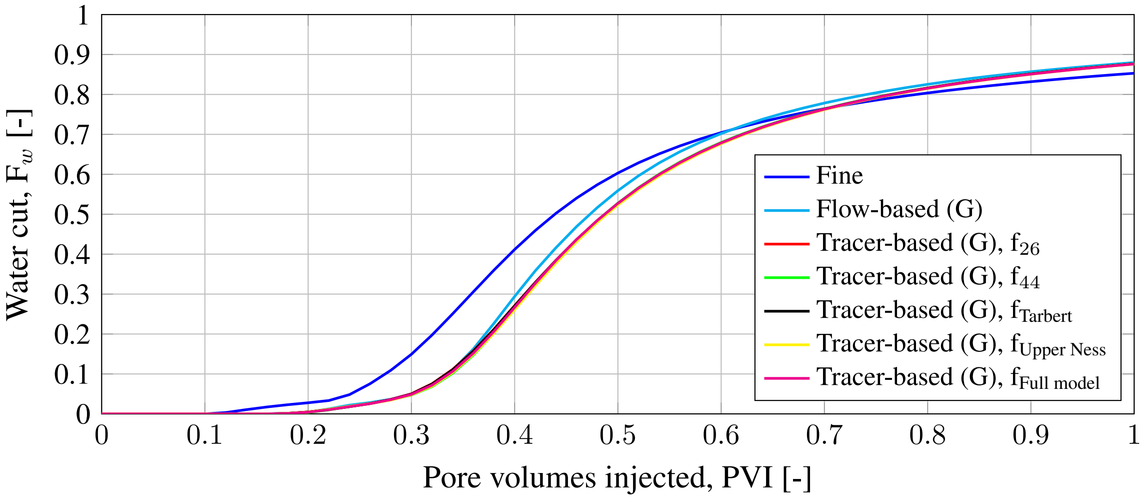


Figure 4.15: Comparison of water cut when global single-phase upscaling is used for the full model. Note that a number of tracer-based water cut results are not visible since the results from the tracer-based upscaling are almost identical.

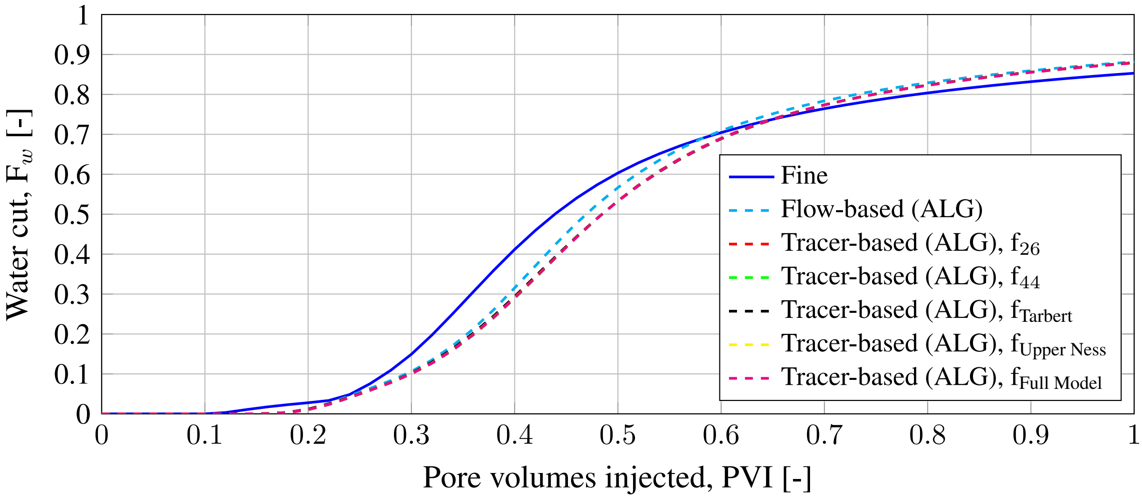


Figure 4.16: Comparison of water cut when ALG single-phase upscaling is used for the full model.

	Layer 26		Layer 44		Full model	
Method	Global	ALG	Global	ALG	Global	ALG
Flow-based	3.91 %	3.85 %	3.06 %	10.7 %	10.5 %	7.62 %
Tracer-based, f_{26}	3.23 %	3.17 %	3.94 %	12.2 %	11.5 %	9.90 %
Tracer-based, f_{44}	3.20 %	3.16 %	4.02 %	12.3 %	11.8 %	10.1 %
Tracer-based, f_{Tarbert}	3.22 %	3.17 %	3.94 %	12.3 %	11.5 %	9.80 %
Tracer-based, $f_{\text{Upper Ness}}$	3.18 %	3.13 %	4.21 %	12.4 %	12.1 %	10.0 %
Tracer-based, $f_{\text{Full model}}$	3.19 %	3.14 %	4.06 %	12.4 %	12.7 %	10.0 %
Tracer-based, f_{Norne}	2.69 %	2.64 %	4.90 %	12.9 %	12.1 %	10.4 %
Tracer-based, f_{SAIGUP}	2.58 %	2.50 %	5.40 %	13.3 %	13.2 %	10.4 %

Table 4.3: Relative error between fine-scale and upscaled water cut results. The keywords "Global" and "ALG" indicates which single-phase upscaling procedure that was used.

CHAPTER 5

DISCUSSION

A discussion and comparison of the results are included in this chapter. Like the previous chapter, this chapter is divided into discussions of the results for single layers and the full model. It is important to keep in mind that the aim of the tracer-based upscaling is to produce as accurate upscaled results as possible while having the same underlying single-phase properties with associated errors as the flow-based upscaling. A comparison between the upscaling procedures is therefore more relevant than comparing the results from the upscaled models with the fine-scale results. However, accuracy with respect to the fine-scale model is important for an upscaled model and reasons why or why not the upscaled results are accurate will be included.

5.1 Single layers

The results for the single layers may be found in Chapter 4 Section 4.1. The difference between the water cuts is used as a quantitative measure of the accuracy of the upscaling methods compared to the fine-scale solution and each other, while the saturation visualizations in Appendix A are used as an additional measure of the quality.

As previously mentioned, the single layers were chosen as the layers with the outermost functions in the stack in Fig. 4.2 to be able to investigate the dependency of the results on the tracer function. This makes an interesting comparison since Fig. 4.2 shows some separation between the functions in the Tarbert and Upper Ness formation. It was therefore expected that the results obtained by using different tracer functions in the same layer would show some discrepancy with respect to the flow-based upscaling.

As seen in Figs. 4.3 and 4.4 this is not the case. The results obtained with the tracer-based method is very close to the results from the flow-based method for both tracer functions. With the scale of the figures, it is almost impossible to tell them apart. **Fig. 5.1** shows an enlarged part of the water cut curve for layer 26. The results obtained by using both tracer functions are close to the flow-based results, and the results from the tracer-based upscaling are in fact closer to the fine-scale solution than the results from the flow-based upscaling. As there is some separation between the functions in the Upper Ness and Tarbert formation, this indicates that the upscaled results are not too sensitive to the tracer function.

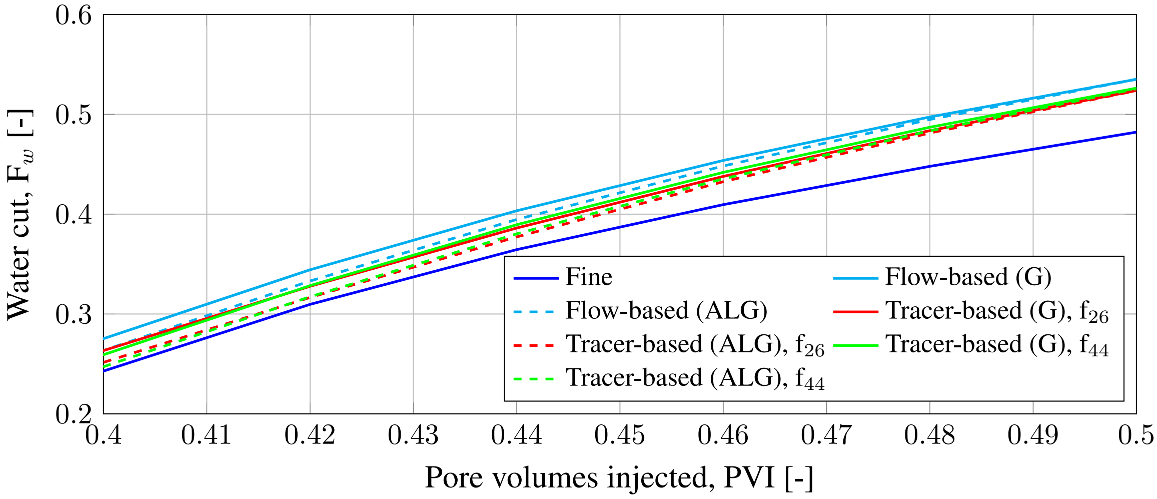
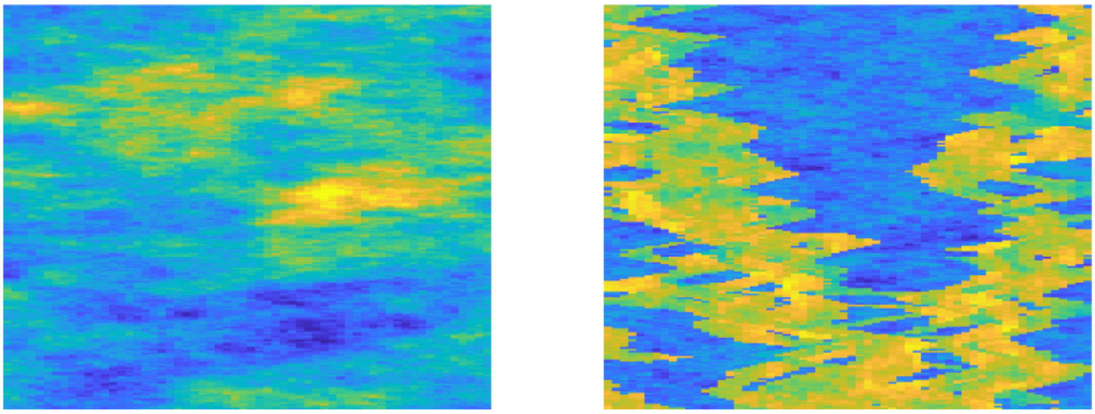


Figure 5.1: Enlarged water cut curve for layer 26.

The saturation visualizations in Fig. A.2 and Fig. A.3 also shows the accuracy of the tracer-based upscaling compared to the flow-based upscaling. For both layers 26 and 44, the saturation distributions obtained from the tracer-based upscaling are almost identical to those obtained from the flow-based upscaling. Also notable from the saturation distributions is that most of the flow in layer 44 occurs in channels, while layer 26 has a more even displacement. This may be seen by comparing the horizontal permeability of the layers shown in **Fig. 5.2** with the saturation distributions in Figs. A.2 and A.3.

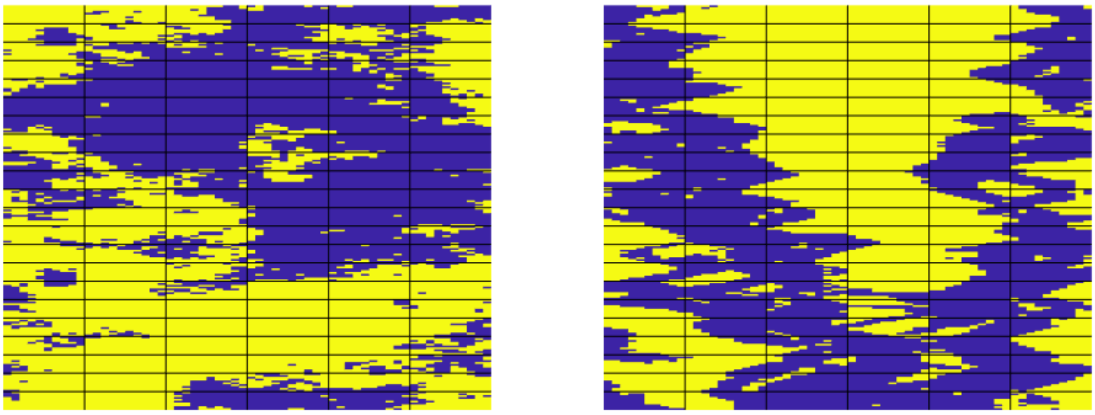

(a) Layer 26

(b) Layer 44

Figure 5.2: Horizontal permeability of the single layers used in simulations.

The fact that most of the flow occurs in channels in layer 44 may also be seen in Fig. 4.6, which shows the upscaled flow-based relative permeability curves of layer 44 compared to the fine-scale relative permeability curves. The channels in the layer will correspond to fine-scale region 1, since it has a higher absolute permeability than region 2. Fig. 4.6 is plotted with opacity, such that a strong black color will indicate that a lot of upscaled relative permeability curves are similar. The strongest black color is closer to the original curves of region 1 than region 2 and thus more flow goes through region 1. The relative permeability curves of layer 26 shown in Fig. 4.5 indicate a more even displacement, which coincides with Fig. A.2 and

Fig. 5.2a. These observations will of course not be valid if the original domain is changed. A domain containing only cells from region 2 could as an example produce similar upscaled relative permeability curves. **Fig. 5.4** shows that the observation of more flow going through region 1 is correct in this case however.



(a) Layer 26

(b) Layer 44

Figure 5.3: Visualization showing the regions in the coarse grid. Region 1 is shown in blue and region 2 in yellow.

Since the majority of the flow in layer 44 goes through region 1 as shown in Fig. 5.4, accurate upscaled results may be obtained by only performing a single-phase upscaling and using the fine-scale relative permeability curves of region 1 directly on the coarse grid. Doing the same for layer 26 will not give satisfactory upscaled results since the difference in flow between the regions are smaller. Two-phase upscaling is therefore required to obtain accurate upscaled results in layer 26. By using a local two-phase upscaling procedure the upscaled relative permeability curves will correct the flow according to which region that is the dominant region within the block.

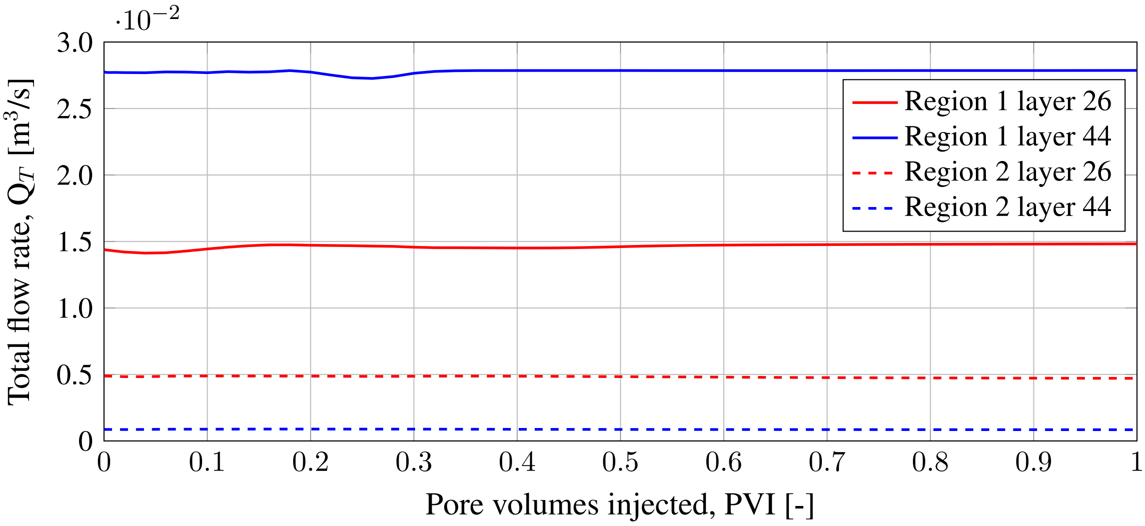


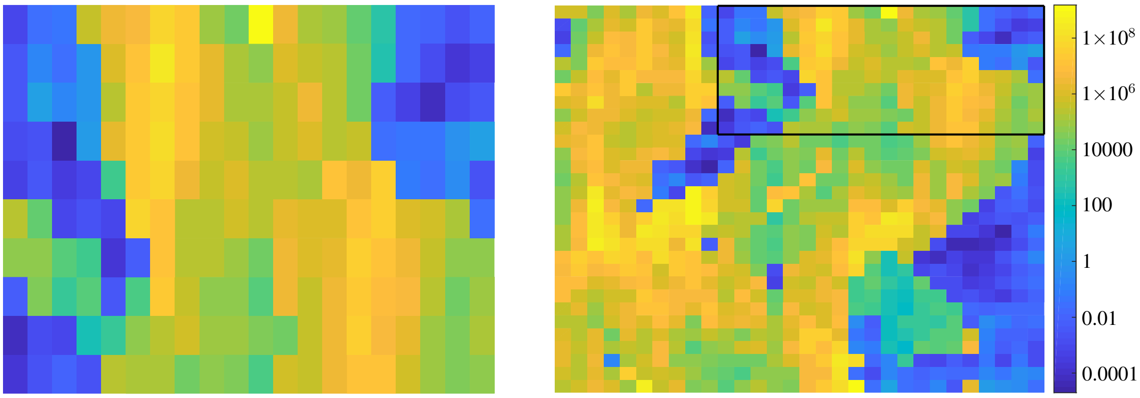
Figure 5.4: Total flow rate through the regions for layer 26 and 44.

A problem often encountered in two-phase upscaling when using the standard local boundary conditions is an overestimation of the two-phase flow functions. Chen and Li (2009) explains the overestimation of the two-phase flow functions by considering a layered system on the size of a coarse grid block. When a constant pressure drop is specified across the system, the inlet and outlet flux in each layer will be proportional to the local permeability in the inlet and outlet cells of the layer. Wallstrom et al. (2002a) found that this is the case only for small permeability values and that the local flux converges to a constant value when the permeability is large in the inlet and outlet cells compared to the rest of the system. The proportional relationship resulting from applying the standard local boundary conditions will, therefore, overestimate the local fluxes which lead to overestimated two-phase flow functions and upscaled predictions.

An example of such an overestimation upscaled predictions may be seen in Fig. 5.1. Fig. A.2 and Fig. A.3 also show areas where the saturation has been overestimated for the upscaled models. The saturation has not been overestimated significantly, but some areas that are not subjected to flow in the fine-scale model will see flow in the upscaled models.

The accuracy of the upscaled results compared to the fine-scale results may thus be improved by using different boundary conditions in the local upscaling procedures. See Odsæter et al. (2015), Wallstrom et al. (2002a), Wallstrom et al. (2002b), Chen and Li (2009), Chen et al. (2013), and Chen and Durlofsky (2006b) for discussions of different boundary conditions, and how these improve the upscaled results compared to the fine-scale results. For single layers, the computational cost of the tracer-based upscaling is so low that a global upscaling procedure may, in fact, be used. This will eliminate the problem of assumed boundary conditions and most likely improve the accuracy of the upscaled results.

The accuracy of a global upscaling procedure may be seen in Fig. 4.4 and Table. 4.3. The ALG upscaling of single-phase flow parameters in layer 44 lead to significant errors compared to the fine-scale solution for all upscaled models. This is most likely due to the initial estimates of the upscaled transmissibility. The initial estimates for the upscaled transmissibility in the ALG procedure were found from a purely local upscaling procedure with imposed boundary conditions. Due to channelized flow in layer 44, capturing global effects such as communication between neighboring grid blocks or large scale flow patterns is important. **Fig. 5.5** shows the difference between a purely local and extended-local domain. Notice how the extended-local domain in Fig. 5.5b reveals that the high-permeability zone in the local domain is part of a larger system.



(a) Purely local

(b) Extended-local

Figure 5.5: Comparison of permeability in units mD in a purely local and extended-local solution domain in layer 44. The purely local domain in (a) is shown in (b) with thick lines.

Since the initial estimates are found on the domain shown in Fig. 5.5a, they will not include any of the effects of the larger system. An example is that the local domain will have more flow in the low permeable area than the extended-local domain, which will deviate from the global flow pattern since most of the flow occurs in the high permeable zones. The resulting error may be seen in **Fig. 5.6** which shows that the initial estimates give flow in areas where the estimates from the ALG procedure and the fine-scale solution do not. If the initial estimates are used in conjunction with the flow-based upscaling, the error is 36.3 % in layer 44. Even though the re-calculation procedure improves the accuracy of the upscaled transmissibility, the initial estimates lack too much accuracy in order to give results close to the fine-scale results under the given constraints.

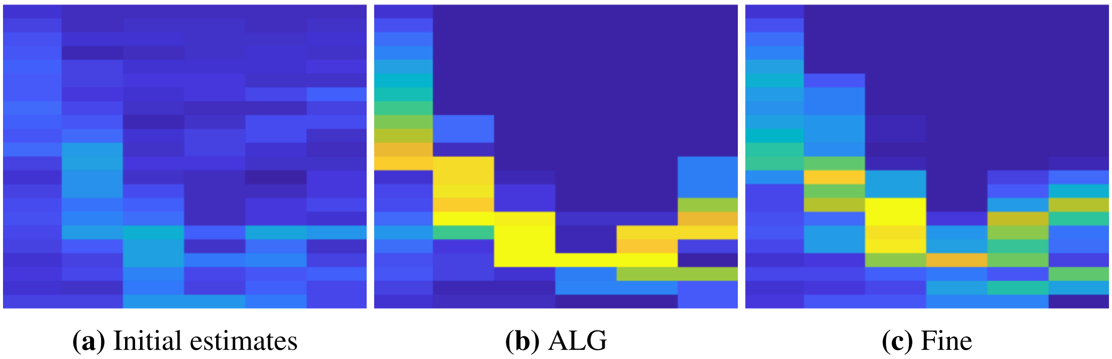

(**a**) Initial estimates (**b**) ALG (**c**) Fine

Figure 5.6: Comparison of flow in layer 44 showing that the initial estimates for the upscaled transmissibility overestimates the velocity in areas of low flow. The figures are plotted on the color scale shown in Fig. A.1.

The initial estimates effect on the upscaled results is demonstrated in **Table. 5.1**. The table shows results obtained by varying the threshold and the criteria on pressure and flux change from one iteration to another when the initial estimates are made with a purely local and extended-local method. The variations give improvements in the results when a purely local domain is used, but not as much as using an extended-local domain. The extended-local domain corresponds to the neighboring coarse blocks of the target coarse blocks.

Table. 5.1 also show that the threshold has the most influence on the accuracy of the ALG single-phase upscaling. When the criteria for pressure and flux change are changed, the error remains the same while the CPU-time increases. This is due to the fact that convergence usually is obtained within the first few iterations regardless of the value of the threshold. The reason why the threshold has the most influence on accuracy is a simple one. The threshold will determine how many of the initial estimates that will be recomputed at each iteration. Thus, if the initial estimates are poor a low threshold should in principle give better results. Table. 5.1 shows that this is not the case however, which calls for a different thresholding procedure.

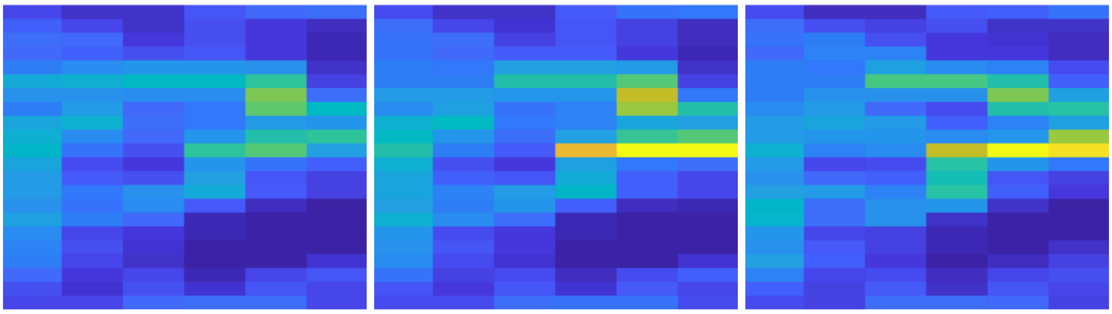
The difference in CPU-time between the purely local and extended-local upscaling of the initial estimates is also marginal for the cases presented in the Table. 5.1. This difference should be expected to grow when the size of the simulation model increases, simply because of the size of the extended-local domain increases.

As shown in Table. 4.1, the global single-phase upscaling procedure is more computationally efficient than the ALG-upscaling. The global procedure is, therefore, the obvious choice for layer 44 when comparing the results in Table. 4.3. Since the computational demand in a local upscaling procedure scales linearly with the number of coarse grid blocks, the ALG upscaling should be more efficient as the simulation model grows larger. The global upscaling procedures do not scale linearly with the number of blocks, which may cause them to become very computationally demanding as the simulation models become large. Further discussions on the computational demand of the methods are given in the next section.

			Purely local		**Extended-local**	
Threshold	β_p	β_Q	**CPU**	**Error**	**CPU**	**Error**
0.10	0.0100	0.0010	1.00	10.7 %	1.87	8.45 %
0.10	0.0001	0.0001	1.16	10.7 %	3.01	8.45 %
0.05	0.0100	0.0010	1.11	8.32 %	1.45	6.90 %
0.05	0.0001	0.0001	2.15	8.32 %	3.23	6.90 %
0.01	0.0100	0.0010	1.75	12.5 %	2.13	11.3 %
0.01	0.0001	0.0001	4.67	12.5 %	6.47	11.3 %

Table 5.1: Table showing how changing control parameters in the ALG single-phase upscaling procedure affects the results from the upscaled model with flow-based two-phase upscaling in layer 44. The CPU-times are normalized by the base-case for purely local upscaling shown on the top line.

For layer 26, the use of ALG single-phase upscaling gives the best results. As previously discussed and shown in Fig. 5.4, the flow is more evenly distributed in the regions in layer 26 than in layer 44. The initial estimates for the upscaled transmissibility will, therefore, give more accurate results with respect to the fine-scale solution as shown in **Fig. 5.7**, even when using a purely local domain. The re-calculation procedure will improve the initial estimates and hence provide the accuracy shown in Table. 4.3.



(a) Initial estimates **(b)** ALG **(c)** Fine

Figure 5.7: Comparison of flow in layer 26 showing that the initial estimates for the upscaled transmissibility accurately models the flow. The figures are plotted on the color scale shown in Fig. A.1.

In addition to showing the robustness of the global single-phase upscaling, the discussion above highlights the importance of accurate upscaled single-phase parameters in two-phase upscaling. The upscaled models obtained from tracer-based two-phase upscaling gives results in close agreement with those obtained from a flow-based upscaling regardless of the single-phase upscaling procedure. This is especially demonstrated in Fig. A.6, where the relative error between the upscaled saturation distributions is plotted versus PVI.

The maximum error between the methods is 1.9 % in layer 26 and 5.9 % in layer 44, while the average errors are 0.9 % and 3.8 %. Note that the errors are found by comparing the results from the flow-based method with the tracer-based results obtained by using the tracer function corresponding to the layer in question and by using global single-phase upscaling for both methods. As shown in Table. 4.3 results from other tracer functions would reduce the error between the methods in layer 44, while other functions or ALG single-phase upscaling would increase the error in layer 26 since the traced-based method is more accurate with respect to the fine-scale solution than the flow-based method. The small errors show the robustness of the tracer-based upscaling, and due to the lower computational cost makes it more desirable to use than the flow-based upscaling for the single layers.

5.2 Full model

The discussions made in the previous section showed that the tracer-based upscaling provides accuracy equal to or better than the flow-based upscaling for single layers. Providing accuracy for single layers may not be sufficient, however, as most reservoir models have a lot more cells and complexity. This section will show that the tracer-based upscaling procedure gives satisfactory accuracy for the full model as well, especially with respect to the results obtained with the flow-based upscaling.

As seen in Fig. 4.7 and Fig. 4.8 the water cut curves from the upscaled model with tracer-based upscaled relative permeabilities follow the curve from the flow-based upscaled model closely. The close relationship may also be seen in Table. 4.3 where the difference from the fine-scale solution is quantified and in the visualizations of the saturation distributions in Fig. A.4. As shown in Fig. A.6, the average error between the saturation distributions of the upscaled models is only 2.6 %.

Thus as for the single layers, the tracer-based upscaling gives results very close to those of the flow-based upscaling at a much lower computational cost as seen in Table. 4.2. Note that since the computational demand for the full model is much larger than for the single layers, the normalized values for the full model should not be confused with the values for the single layers in Table. 4.1.

Another similarity with the results for the single layers is the fact that the ALG single-phase upscaling is much more computationally demanding than the global single-phase upscaling. By using the constraints described in Section 3.2.2 the ALG upscaling is 116 times slower than the global upscaling, while one iteration in the re-calculation procedure is 11 times slower. The CPU-times used in these comparisons were obtained by looking at only the single-phase upscaling procedure and should therefore not be confused with the CPU-times in Table. 4.2.

As described in the previous section, the main control parameter for the computational cost of the ALG procedure is the threshold. Due to the large cost of performing the ALG upscaling, it is therefore of interest to investigate if the threshold can be increased without effecting the accuracy of the results. **Table. 5.2** shows that a lower computational cost is obtained at the expense of accuracy.

It was stated in the previous section that the global upscaling procedure should become more computationally expensive for the full model. Both Table. 4.2 and 5.2 shows that this is not the case however, which may be explained by considering the nature of the global upscaling procedure. Since single-phase parameters are upscaled, the cost of a global pressure solution is much less demanding than for two-phase flow. In addition, since no iterations are performed only one pressure solution with flow in each principal direction is needed. As described in Chapter

2, anomalous or negative upscaled transmissibilities often occur when iterations are not performed. If such upscaled transmissibilities occur, they are simply replaced by a transmissibility calculated by using upscaled absolute permeabilities and equation (2.14).

Threshold	β_p	β_Q	**CPU**	**Error**
0.10	0.0100	0.0010	116.3	7.62 %
0.20	0.0100	0.0010	26.2	10.7 %
0.30	0.0100	0.0010	11.3	11.4 %
0.40	0.0100	0.0010	7.1	12.1 %
0.50	0.0100	0.0010	3.2	13.4 %
0.60	0.0100	0.0010	1.3	13.9 %
0.70	0.0100	0.0010	0.9	14.0 %
0.80	0.0100	0.0010	0.6	14.2 %
0.90	0.0100	0.0010	0.3	14.3 %

Table 5.2: Table showing how changing the threshold affects the CPU and accuracy when the ALG single-phase upscaling in conjunction with the flow-based two-phase upscaling for the full model. The CPU-times are normalized by the global single-phase upscaling.

Another key to the efficiency of the global upscaling procedure is a highly efficient linear solver which use an aggregation-based algebraic multigrid method (Notay, 2019). Despite of the efficiency, the use of global single-phase upscaling only gives the best results for layer 44. For layer 26 and the full model, the ALG single-phase upscaling gives the best results as shown in Table. 4.3. Since the tracer-based two-phase upscaling gives results close to those of the flow-based method at 3 times lower computational cost on average, it may be used in conjunction with the ALG single-phase upscaling and still be more efficient than the flow-based two-phase upscaling with globally upscaled single-phase parameters as shown in Table. 4.2.

The efficiency of the tracer-based upscaling may be better illustrated with actual CPU-times rather than normalized ones. The CPU requirements for single-phase upscaling procedures are minimal compared to two-phase upscaling procedures. As previously stated the global single-phase upscaling was 116 times faster than the ALG upscaling, which equals 0.8 hours in actual time. Even though the tracer-based upscaling is only three times faster than the flow-based upscaling, this equals 16.5 hours in actual time. The simulation times were obtained by using a *Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz, 3301 Mhz, 4 Cores, 8 Logical Processors* processor. The differences in time motivate the use of ALG and tracer-based upscaling rather than flow-based upscaling.

A difference between the two-phase upscaling procedures is the resulting upscaled relative permeability curves shown in Fig. 4.9 and 4.10. The curves from the flow-based method show the typical behavior of upscaled relative permeability curves with outliers, while the ones from the tracer-based method tend to smoothen out more as the water saturation approaches 0.6. This may be explained by the methodology described in Section 3.2.2 which is used to create the tracer function. Since a binned average is used to create the functions, the functions will flatten out as shown in Fig. 4.14. A large difference in tracer saturation will therefore not give a large difference in water saturation. An example is that the function for the full model will give a water saturation of 0.5-0.6 for tracer saturations in the interval 0.2-0.8.

As seen in Fig. 4.10, the tracer-based relative permeability curves smoothen out and follow the fine-scale curves at a water saturation around 0.5 and 0.6. The reason for this is thus that the tracer function rarely gives water saturations above 0.6 due to the shape of the function and the relative permeability curves are interpolated to the endpoint. **Fig. 5.8** shows that the same happens when the function from the Saigup model is used. This may thus be a possible source of error for the tracer-based upscaling and explain why the flow-based upscaling gives better water cut results for the full model.

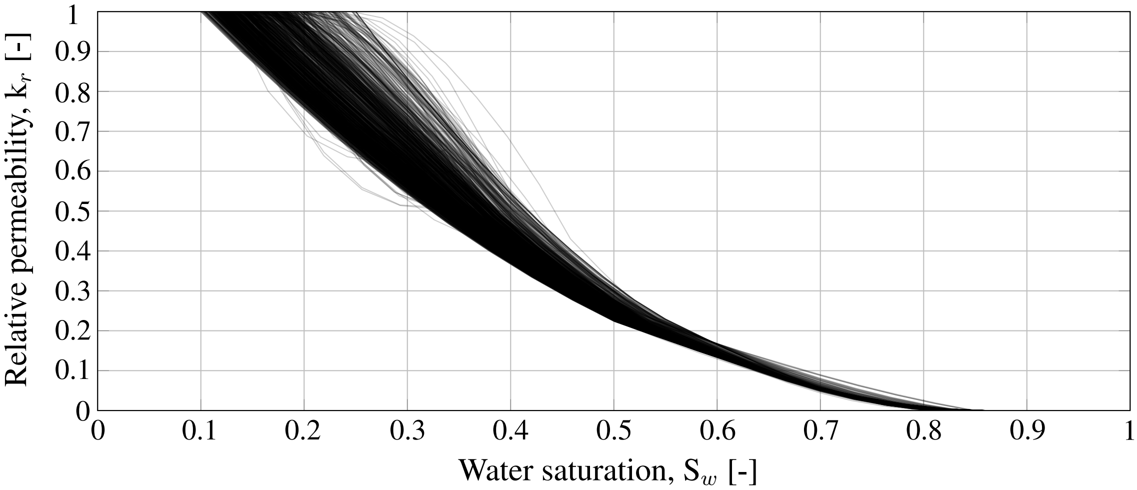


Figure 5.8: Tracer-based upscaled relative permeability of oil in the full model with the function f_{Saigup}

The upscaled models' accuracy with respect to the fine-scale model is reasonably good. However, both the flow-based and tracer-based upscaling give a late breakthrough which is typically encountered when the upscaled models include no two-phase upscaling. The late breakthrough may thus indicate that the quality of the upscaled models' transport predictions are somewhat inaccurate for the full model. The errors associated with the total flow rate are quite small as shown in Figs. 4.12 and 4.13 however, which may be attributed to the single-phase upscaling procedures. The single-phase flow parameters often play an important role in the flow rate results for two-phase flow, and the single-phase upscaling procedures used here are among the most accurate.

There is a difference in the results depending on which single-phase upscaling procedure that is used, however, which indicates that the quality of the upscaled models will improve if the upscaled single-phase parameters are improved. Any errors resulting from the single-phase upscaling will have a greater effect for the full model than for the single layers, such that some error accumulation from the upscaled single-phase flow parameters should be expected. Another possible reason for the late breakthrough is that of numerical dispersion due to the coarser discretization, since steady-state methods do not compensate for such (Pickup et al., 2000). A test to verify or discard the possibility of numerical dispersion could be made by performing simulations a fine grid model where the parameters are taken from the coarse model. This was not done in this work, but some dispersion should be expected since the grid is coarsened by a degree of 10 in each principal direction.

A source of error for the upscaled results in the full model is the fact that the upscaled relative permeability is only found with flow in the x-direction. As seen in **Fig. 5.9**, the upscaled relative permeability curves in the z-direction are considerably different from the curves in the x- and y-direction. For the single layers, the differences in relative permeability curves will not have a large impact on the results since no flow occurs in the z-direction and since the upscaled relative permeability curves in the x- and y-direction are almost identical. In the full model, flow will occur between the layers and the upscaled relative permeability curves obtained from flow in the x-direction will therefore not give correct transport predictions. Expressing the upscaled relative permeability as a tensor with values for each principal direction would thus most likely produce more accurate results for the full model.

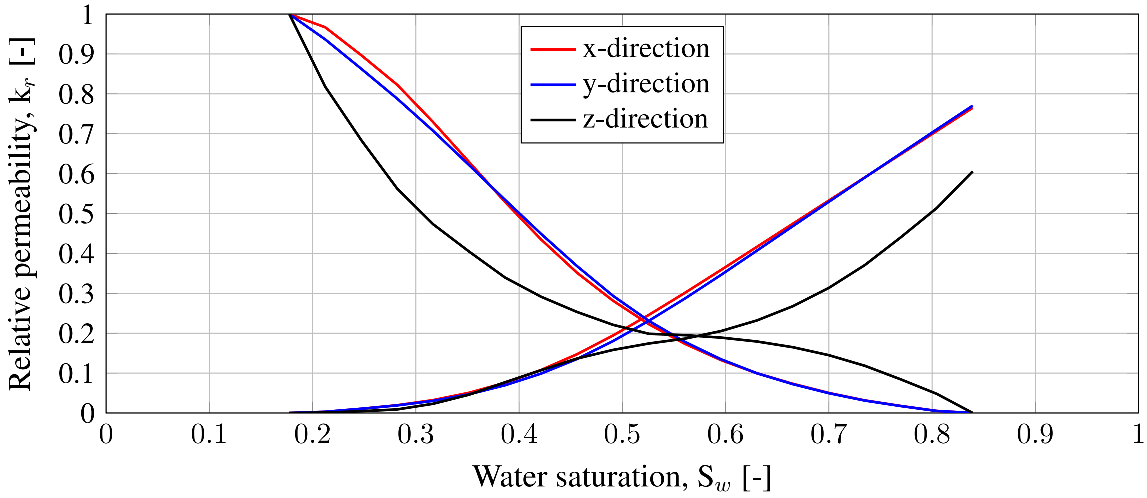


Figure 5.9: Upscaled relative permeability curves in each principal direction for a coarse block in the full model.

It was mentioned in Chapter 3 that capillary pressure was introduced to the models when the fluid structure was created by the *getExampleFluidProps* function in MRST. The addition of the capillary pressure was unintentional, and the effect of neglecting it for both the fine- and coarse-scale models should therefore be investigated. For this reason, Fig. 4.11 shows results with and without capillary pressure for the full model. The figure shows that the effect of the capillary pressure on the results is negligible for the coarse models, which is most likely due to the scale of the coarse grid blocks, i.e. the capillary forces will be dominated by the viscous forces. Neglecting the capillary pressure also increases the accuracy of the upscaled models with respect to the fine-scale model at high PVI.

As a concluding remark on the results for the full model, a reminder that the purpose of this work is to compare the accuracy of the tracer- and flow-based upscaling procedures is given. The errors described above are present for both procedures, but the differences between the upscaling procedures are small. The results for the full model may thus be considered satisfactory.

5.3 Sensitivity to the tracer function

As discussed in section 5.1, the upscaled tracer-based results were not as sensitive to the tracer function as first expected. It is therefore of interest to investigate the effect of using different tracer functions further. The comparisons in the sections above were based on results obtained from using tracer functions generated within the same model. In order to properly investigate the tracer function's effect on the results, functions from different models should be used and compared as shown in Fig. 4.14.

Table. 4.3 shows the relative error between the fine-scale results and the different upscaled results. The table, in fact, show that the best results for the tracer-based upscaling do not occur when using the tracer function corresponding to the single layers or the full model. Worth noting is that the functions from the SAIGUP and Norne models give the best results in layer 26 and reasonable accuracy in layer 44 and the full model.

The fact that the functions from the SAIGUP and Norne models give more accurate results indicates that the function's impact on the results is more dependent upon the shape of the function rather than the model the function is generated from. As seen in Fig. 4.14, the functions from the Saigup and Norne models have different shapes than those generated from the SPE10 model which has equal or similar shapes. The importance of the shape of the function was also mentioned in the previous section when the smoothness of the upscaled tracer-based relative permeability curves was discussed.

Since the shape of the function will determine the water saturation distribution, it is important that the function is shaped such that the distribution is as close to the actual as possible to obtain accurate upscaled relative permeability curves. As previously mentioned and exemplified through the fact that the function for the full model will give a water saturation of 0.5-0.6 for tracer saturations in the interval 0.2-0.8, a large span in tracer saturation values gives a narrow interval of water saturations. Since this might give inaccurate upscaled relative permeability curves, new ways of creating the tracer function where the resulting shape is different should be considered.

The tracer functions used here were created in a very simplistic way by simply binning and averaging data from multiple time steps. One possible source of error by doing so is shown in **Fig. 5.10**, where the development of the tracer function for layer 26 with PVI is shown. The figure shows that the shape of the function changes with the amount of injected fluid, which indicates that time dependency ideally should be introduced to the tracer function. The time-dependency in Fig. 5.10 is incorporated in the plot through the colorbar shown below the figure. Thus, a function displayed in blue color will correspond to a low value of PVI while orange and yellow functions correspond to a high value of PVI.

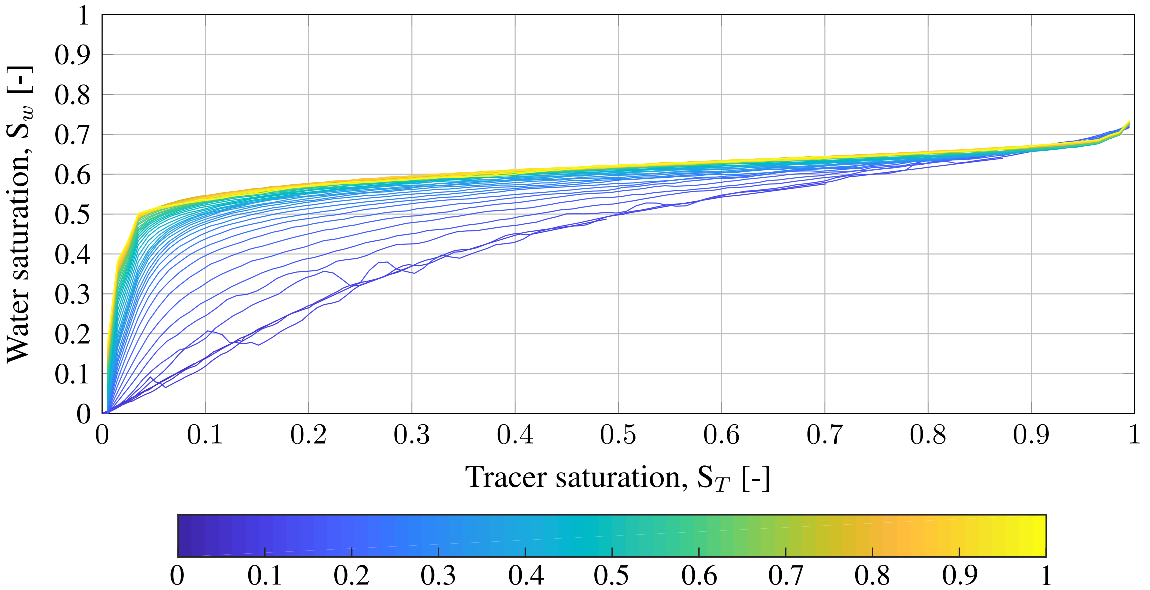


Figure 5.10: Time dependent tracer function for layer 26. The development with time is incorporated in the figure by using the color scale shown in the color bar to distinguish functions at different PVI.

Another possible source of error for the tracer-based method is the scale discrepancy of the tracer function. The tracer functions are generated based on regular and tracer simulations on the fine scale, but they are used to generate results on the coarse scale. In order to investigate this, tracer simulations were performed on both the coarse and fine scale for the full model. **Fig. 5.11** shows a cross plot of the tracer saturations for all time steps, where the x-axis is the coarse scale tracer saturations and the y-axis is the pore volume averaged fine-scale tracer saturations.

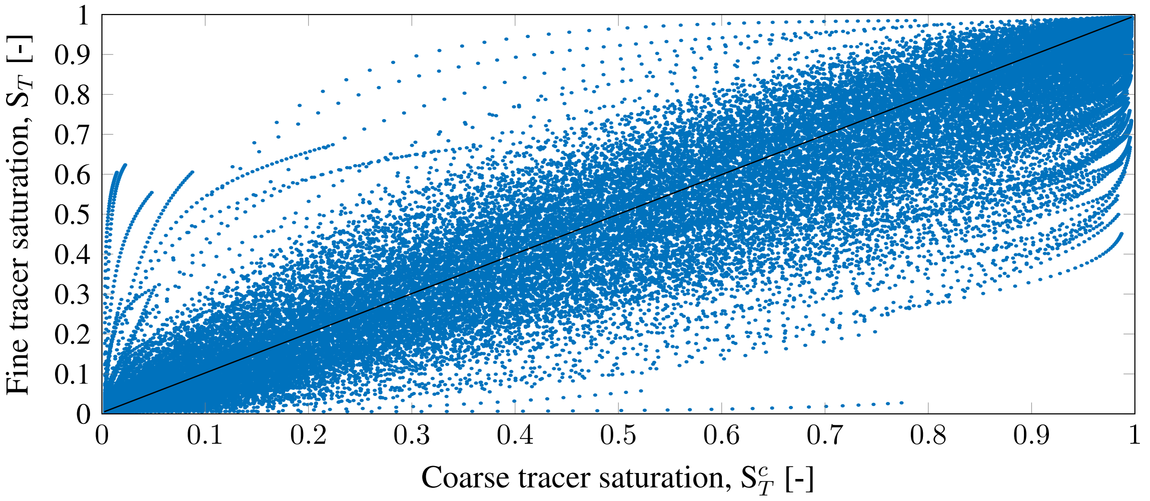


Figure 5.11: Cross plot of coarse- and fine-scale tracer saturations.

Fig. 5.11 shows that there is not a perfect match between the tracer saturations on the fine and coarse scale. The result of this is shown in **Fig. 5.12**, where tracer functions generated from fine- and coarse-scale simulations are compared for layer 26 and 44. The functions originating from the coarse scale are denoted by the superscript c. The coarse-scale tracer functions are noticeably different from the fine-scale functions in that they cover a larger range of saturation values through a different shape.

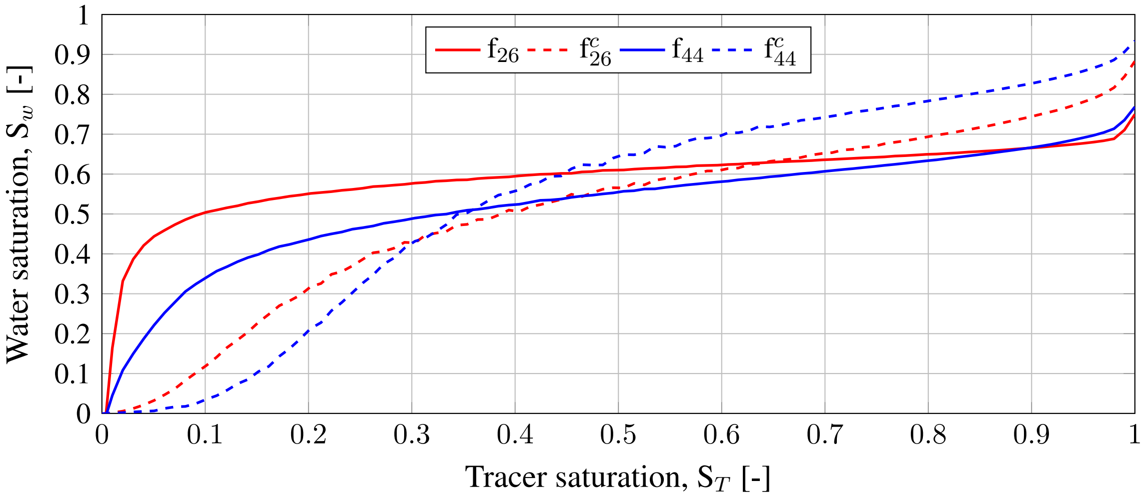


Figure 5.12: Comparison of tracer functions generated on the fine and coarse scale for layer 26 and 44.

Even though the functions generated on the coarse scale have different shapes than those generated at the fine scale, significant improvements were only obtained in layer 44 when using coarse-scale tracer functions. This again highlights the fact that the shape of the function is crucial for obtaining accurate results.

A similarity between the fine- and coarse-scale tracer functions is the development with PVI as shown in Fig. 5.10 and **Fig. 5.13**, where the time-dependent coarse scale tracer function for layer 26 is displayed. Both the functions generated on the coarse scale and fine scale will give larger water saturations for a particular tracer saturation as the number of PVI increases.

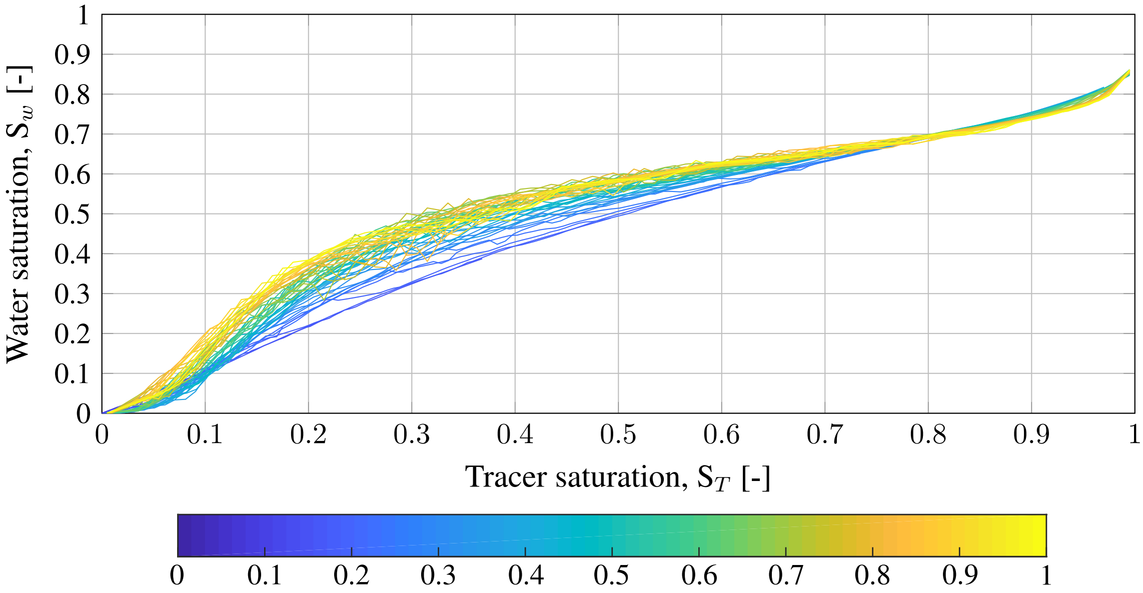
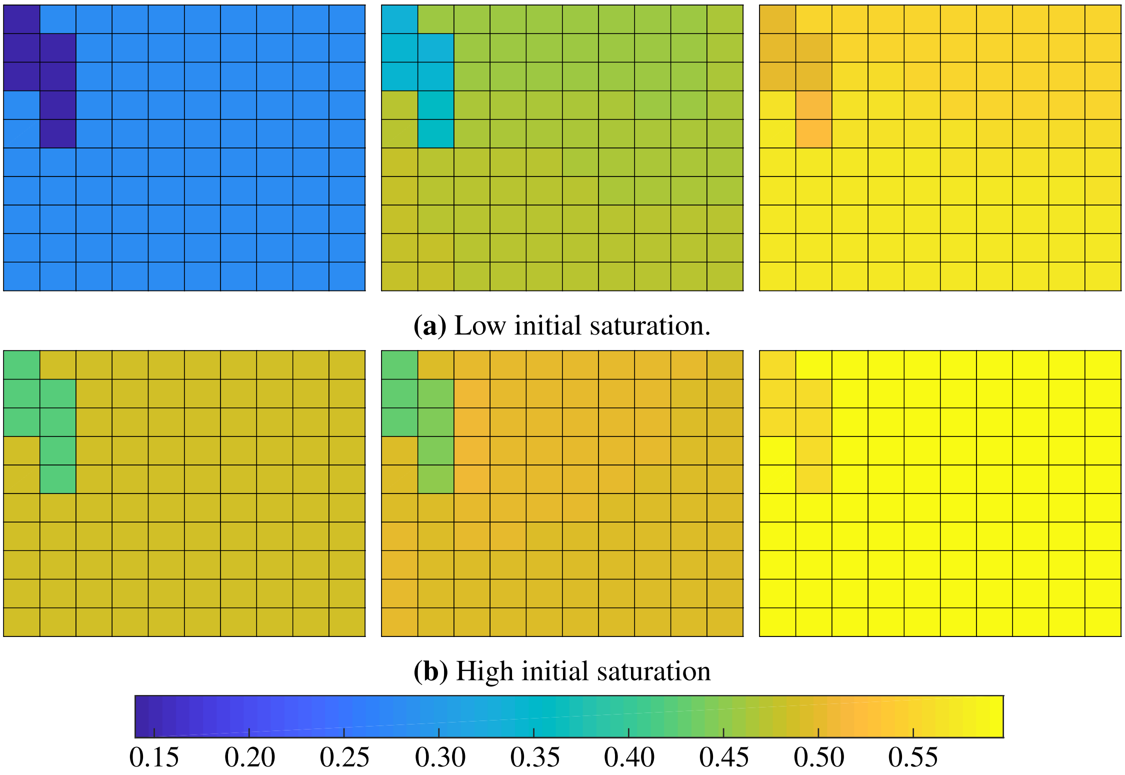


Figure 5.13: Coarse scale time dependent tracer function for layer 26. The development with time is incorporated in the figure by using the color scale shown in the color bar to distinguish functions at different PVI.

Another way of thinking of PVI is the amount of injected fluid present in the model. As the number of PVI of e.g. water increases, the amount of water present in the model will increase. The total water saturation within a grid block will influence the steady-state simulations made in the upscaling procedures. If the magnitudes of the initial saturation distribution are low, the difference between the steady-state distribution and the initial distribution will be large as shown in **Fig. 5.14a**. On the other hand if the magnitudes of the initial saturation distribution are high, the steady-state simulations will not increase the magnitudes by much as shown in Fig. 5.14b.

Fig. 5.14 shows saturation distributions for a block in layer 26 when a function corresponding to the layer generated on the fine scale is used in the tracer-based method. As seen in the right plot in Fig. 5.14a, the saturations are overestimated in the tracer-based method when the initial saturations are low. Also noticeable is that the difference between the saturation distributions obtained by using the tracer function is small. This overestimation and low difference is an example of the error obtained by using an averaged function for all time steps.



(a) Low initial saturation.

(b) High initial saturation

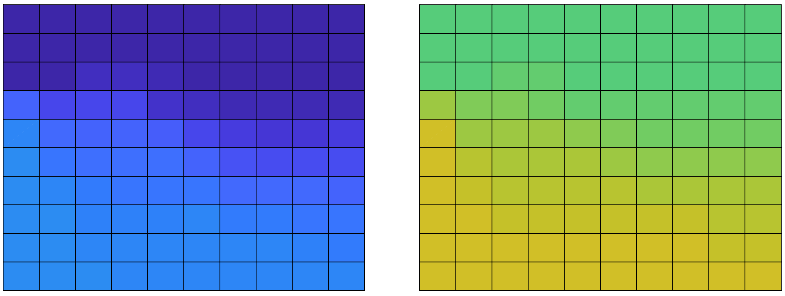
(c) Color scale

Figure 5.14: Comparison of initial saturation distribution (left) and the steady-state actual saturation distributions in the flow-based (middle), and tracer-based (right) methods for a grid block in layer 26.

If time dependency and the correct shape, illustrated in Figs. 5.10 and 5.13, is introduced to the tracer function, the steady-state saturation distribution obtained by using the tracer function will be more equal to the distribution obtained in the flow-based method. This may be seen by comparing the steady-state tracer saturation distributions shown in **Fig. 5.15** with the time-dependent functions.

Consider Fig. 5.15a and the time-dependent function in Fig. 5.10. The steady-state tracer saturation distribution corresponds to a low value of PVI such that one of the functions in blue color would be appropriate to use. Since the tracer saturation distribution in Fig. 5.15a has values of approximately 0.15-0.25, the lowermost functions in blue would give an actual saturation distribution equal to that in the middle figure in Fig. 5.14a. On the other hand, when using the function generated by the averaging procedure shown in yellow color in Fig. 5.10 the saturations are overestimated as shown in the right figure of Fig. 5.14a.

When the initial tracer saturation values are high, which corresponds to a large value of PVI, the difference between the functions in Fig. 5.10 are smaller and reasonable accuracy with respect to the actual saturation distribution is obtained when using the function generated by the averaging procedure as shown in Fig. 5.14b. Improvement should be expected for this case as well if the function corresponding to the number of PVI is used however.



(a) Low initial saturation. **(b)** High initial saturation

Figure 5.15: Steady-state tracer saturation distributions for a block in layer 26 on the color scale shown in Fig. 5.14c.

Since the tracer functions generated at the coarse scale produces results in close agreement with those generated at the fine scale, it will be advantageous to further investigate the effect of using coarse-scale functions considering the computational demand. The purpose of performing an upscaling is to avoid multiple full fine-scale two-phase simulations, and generating the tracer function based on such simulations is therefore contradictory. Generating the tracer functions based on coarse-scale simulations will lower the computational cost significantly as demonstrated in Tables. 4.1 and 4.2.

Even though the discussion above has highlighted sources of error for the tracer-based method, the errors do not cause significant discrepancies between the flow-based and tracer-based upscaled results. Thus as previously mentioned, the tracer-based upscaling offers accuracy similar or equal to that of the flow-based upscaling at a much lower computational cost. The additional error by using the tracer-based method may almost be considered insignificant when comparing the computational demand of the methods. In addition if the changes to the tracer function generation procedure described above, like including time dependency or improving the averaging procedure, are made one should expect that the tracer-based results will improve. The results obtained by using different tracer functions are also very similar regardless of the fact that the functions are quite different. This indicates that the tracer-based upscaling is not too sensitive to the tracer function.

5.4 Modelling two-phase flow by single-phase flow

As mentioned in the chapter introduction, the comparison between the tracer- and flow-based upscaling procedures are made based on the fact that they have the same underlying single-phase parameters and their associated errors. The results and previous discussions may thus be considered as investigations on the possibility of modelling two-phase flow by single-phase flow, since the injection of a tracer may essentially be considered as single-phase flow.

The accuracy of the results from the tracer-based method may thus indicate that single- and two-phase flow have similarities with respect to the flow behaviour in a reservoir. It should be noted that this observation is based on the fact that the upscaled relative permeability curves from the tracer- and flow-based upscaling provide the basis of comparison. It is therefore important that inaccurate upscaled relative permeabilities give inaccurate simulation results.

As seen in **Figs. 5.16** and **5.17**, the most accurate upscaled results are obtained when using the fine-scale relative permeability curves of region 1 directly on the coarse grid. This is unfortunate since it indicates that the effect of the upscaled relative permeability for the model used here is small. This in turn might indicate that accurate tracer-based results would be obtained regardless of which tracer function that is used.

Thus, it is not possible to say for sure whether or not the single-phase flow (tracer flow) gives good predictions for the two-phase flow in the model setup used here. The tracer-based upscaling will produce equal or similar results to the flow-based upscaling due to the accuracy obtained by using the fine-scale relative permeability curves of region 1 on the coarse scale. Ideally, the accuracy of the upscaled model should be more dependent upon the quality of the upscaled relative permeability curves. The basis of comparison would then be that accurate upscaled relative permeabilities would be required to obtain accurate upscaled results. This would in turn provide a deeper insight into the ability of the tracer flow to model two-phase flow.

In hindsight, more accurate upscaled results would probably be obtained if only the fine-scale relative permeability curves of region 1 were used in the calculation of the upscaled relative permeability. The figures below show that the relative permeability curves of region 2 gives very poor results, and they will thus worsen the results from the upscaling procedures.

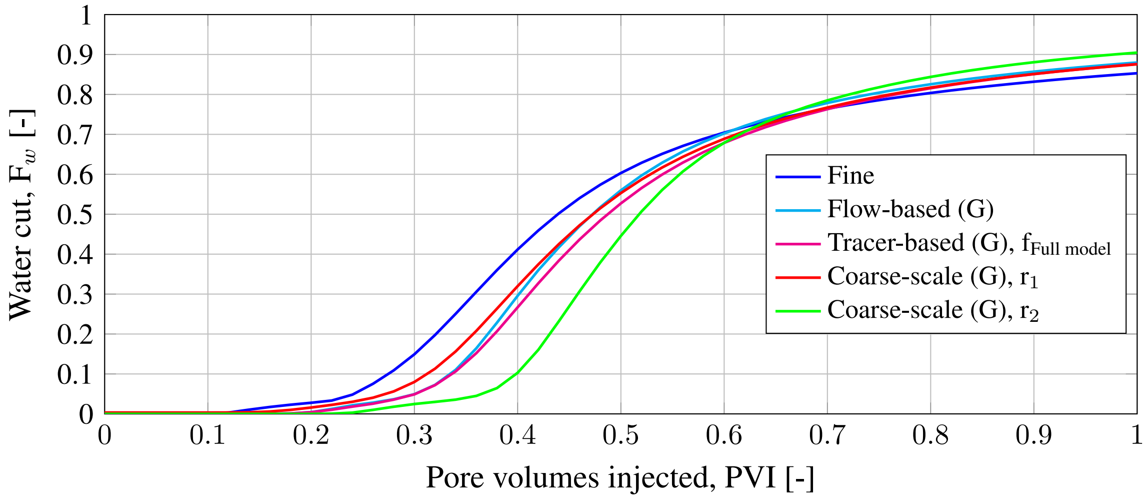


Figure 5.16: Comparison of water cut results for the full model when global single-phase upscaling is used. *Coarse-scale,* r_N are results obtained by using fine-scale relative permeability curves of region N.

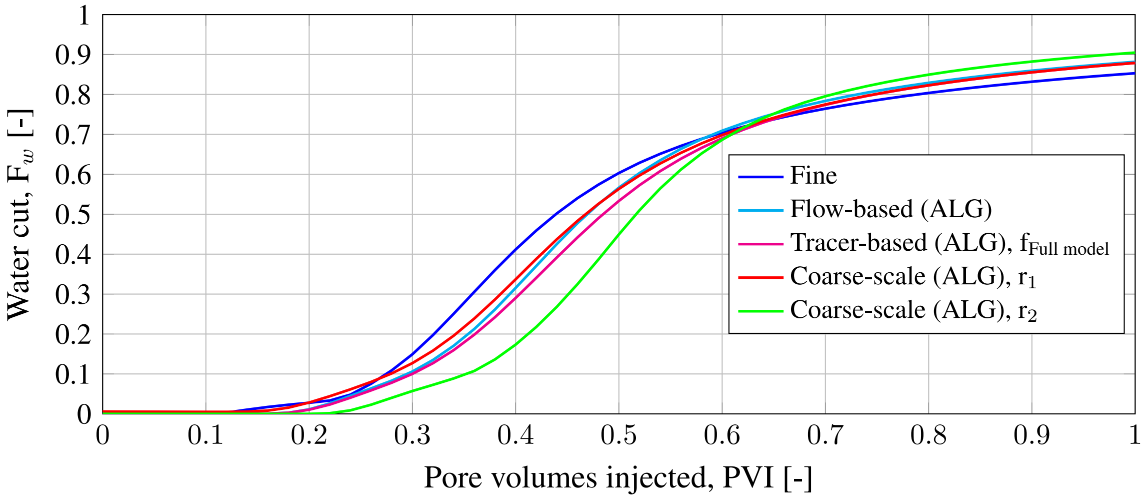


Figure 5.17: Comparison of water cut results for the full model when ALG single-phase upscaling is used. *Coarse-scale,* r_N are results obtained by using fine-scale relative permeability curves of region N.

The previous discussion does not exclude the possibility of modelling two-phase flow by single-phase flow, however. Even though the results presented here gives a poor basis of comparison due to the low dependency on accurate upscaled relative permeability curves, other models could give results that proves that there is a relationship. Simply neglecting the regions or changing the relative permeability curves of the model used here is also a possibility. This is something that definitely should be investigated further.

CHAPTER 6

CONCLUSION

This thesis has presented a tracer-based two-phase flow upscaling approach for the relative permeability, which aims to reduce the computational cost of finding the steady-state saturation distribution in the standard flow-based two-phase upscaling technique. The method uses a predefined function to convert steady-state tracer saturation distributions to actual saturation distributions.

The tracer-based two-phase flow upscaling presented here falls into the category of a local upscaling procedure. The standard local boundary conditions were used, but the method can readily be implemented with any other local boundary conditions. Considering the low computational cost of the method, it may also be used as a global method for small to medium reservoir models.

To investigate the accuracy of the tracer-based upscaling approach, it has been compared with the standard flow-based steady-state upscaling. The fine-scale reference solutions were obtained from both single layers and the full model SPE10. The model was modified such that two regions with different parameters were used to make the upscaling more challenging. Rather than the original five-spot well configuration, generic global flow along the x-direction was considered.

Two highly accurate single-phase upscaling procedures were used to find the upscaled transmissibility in conjunction with the two-phase upscaling procedures. As a continuation of the specialization project, a local-global method was used and compared to a global method. It was shown that the global single-phase upscaling procedure is more computationally effective than the local-global method, despite the fact that global methods usually become very costly as the reservoir model grows larger. This is attributed to the highly efficient linear solver that was used and the fact that no iterations were performed to remove the occurrences of negative and anomalous transmissibilities.

An analysis with respect to the tracer-based upscaling procedure's sensitivity to the tracer function was performed. Functions generated based on data from both the SPE10 model and the Norne and SAIGUP models were used. In addition, results obtained by using functions originating from both fine- and coarse-scale simulations were compared to investigate the possibility of scale-dependency.

The main conclusions that can be drawn from this work are:

- The tracer-based two-phase flow upscaling has shown equal accuracy or errors within a few percent with respect to the flow-based upscaling at a much lower computational cost. This was done by converting steady-state tracer saturation distributions to actual saturation distributions with a predefined function. Possible sources of error for the method has been highlighted, and by addressing these one should expect that even higher accuracy is obtained. This makes the tracer-based upscaling a good alternative to the existing steady-state upscaling procedures, especially in cases where the computational demand is high.
- The results of the tracer-based upscaling were not as dependent upon the tracer function as first expected. The results obtained by using different tracer functions were remarkably close, even though some of the functions were generated based on other models. It was also shown that it would be beneficial to generate the tracer functions based on coarse-scale simulations considering the computational demand and the fact that similar accuracy in the upscaled results was obtained.
- The major source of error for the tracer-based upscaling procedure was proven to be the shape of the tracer function. This indicates that the functions are not model specific as exemplified through the fact that the function from the SAIGUP model gave the best results in layer 26. The fact that the functions change with PVI indicates that some time-dependency should be introduced in the functions. This will also give the functions the correct shape for the given situation and provide increased accuracy in the tracer-based upscaled results.
- The accuracy of the tracer-based method indicates that there are similarities between single- and two-phase flow. It was also shown that these similarities may be due to the fact that accurate upscaled results should be expected due to the small effect of the upscaled relative permeability curves on the results. Thus, in order to investigate the possibility of modelling two-phase flow with single-phase flow, simulations on models where the results are more dependent upon accurate upscaling of the relative permeability must be performed.
- The importance of accurate single-phase upscaling in the modeling of two-phase flow was demonstrated through the fact that different accuracy in the upscaled results was obtained depending on which single-phase upscaling procedure that was used. The robustness of a global upscaling procedure was shown through consecutive accuracy in the results, while the ALG single-phase upscaling was shown to be more dependent upon the heterogeneities in the reservoir.

CHAPTER 7

FURTHER WORK

Throughout the discussion in Chapter 5, several changes to the tracer-based upscaling procedure have been proposed. This chapter will summarize and elaborate on these such that the procedure may be improved and provide even better accuracy both with respect to fine-scale results and other two-phase upscaling procedures.

The most repeated discussion on sources of error for the tracer-based method has been on the shape of the tracer functions that were used in this work. It was shown that the current shape of the tracer function might give inaccurate upscaled relative permeabilities due to the narrow interval of output saturations resulting from the averaging procedure used in the function generation and interpolation procedure. Including time dependency was suggested as a possible solution to this problem, since the shape of the function will be corrected according to the tracer saturation distribution within a coarse grid block. Further investigations on how the shape of the function affects the results are therefore the main recommendation, since the function will have the greatest impact on the accuracy of the upscaled results.

Whether or not there are any other dependencies in the tracer function is also an interesting subject of investigation. One such dependency could originate from the fact that the injection of a tracer will give another sweep efficiency than e.g. the injection of water. The dependencies of the tracer function could be investigated through the use of flow diagnostics on both tracer and regular simulations, which is thoroughly implemented in MRST.

The effect of using tracer functions originating from coarse-scale simulations should be given more attention since this will reduce the computational cost of the tracer generation procedure significantly. The limited testing performed in this work indicates that the results are not affected much by using coarse-scale tracer functions which also highlights the robustness of the tracer-based upscaling for the model setups used here.

The discussions made in Section 5.1 showed that the threshold for which interfaces the transmissibility in the ALG single-phase upscaling procedure is to be re-calculated for had the greatest impact on both the accuracy and computational demand of the method. It was also shown that changing the threshold did not provide the expected improvements in the results. This indicates that a new thresholding procedure ideally should be implemented since accurate single-phase upscaling is often a necessity for obtaining accurate upscaled two-phase flow results.

The tracer-based upscaling should be tested on other different models and compared to other existing two-phase upscaling procedures to ensure its robustness. Performing tests on other models with different global flow scenarios will also give insights into the dependencies of the tracer function. The effect of the relative permeability in the "other" models should ideally be greater than for in the one considered here such that the relationship between single- and two-phase flow may be investigated further.

Due to the low computational cost of the tracer-based upscaling, it might be achievable to implement the method as a global upscaling procedure. As previously mentioned, global two-phase upscaling procedures are rarely performed since global fine-scale solutions of time-dependent flow equations make them very computationally demanding. Comparisons of results and computational cost of a global tracer-based upscaling and existing local upscaling procedures are therefore of interest.

The possibility to use tracer functions as a part of a fine-scale pseudo-simulation could also be investigated. The pseudo-simulations may be thought of as simplified simulations where the tracer fluid is used rather than the actual fluids. By using a tracer function to convert tracer saturations to actual saturations at each time step in a simulation, the computational demand of the fine-scale simulations will be reduced significantly. The reduced computational cost of using tracer simulations would be beneficial if e.g. wells are relocated.

Another interesting subject of investigation is to investigate if the interpolation procedure is necessary in the ALG single-phase upscaling. Rather than interpolating the coarse pressures to the fine-scale boundaries of the local domain, the boundary conditions can be defined such that a pressure condition is only applied at the boundaries which coincides with the coarse pressure locations. No-flow conditions are then applied to the remainder of the boundaries.

REFERENCES

Aarnes, J. E., Kippe, V., Lie, K.-A., 2005. Mixed Multiscale Finite Elements and Streamline Methods for Reservoir Simulation of Large Geomodels. Advances in Water Resources 28 (3), 257–271.

Aarnes, J. E., Kippe, V., Lie, K.-A., Rustad, A. B., 2007. Modelling of Multiscale Structures in Flow Simulations for Petroleum Reservoirs. In: Geometric Modeling, Numerical Simulation, and Optimization. Springer, Ch. 2, pp. 307–360.

Aavatsmark, I., 2002. An Introduction to Multipoint Flux Approximation Methods for Quadrilateral Grids. Computational Geosciences 6 (3-4), 405–432.

Ali, E., Chatzichristhos, C., Aurdal, T., Muller, J., 2000. Tracer Simulation to Improve the Reservoir Model in the Snorre Field. In: SPE International Oil and Gas Conference and Exhibition. Bejing, China.

Amidror, I., 2002. Scattered Data Interpolation Methods for Electronic Imaging Systems: A Survey. Journal of Electronic Imaging 11 (2), 157–176.

Branets, L. V., Ghai, S. S., Lyons, S. L., Wu, X. H., 2008. Challenges and Technologies in Reservoir Modeling. Communications in Computational Physics 6 (1), 1–23.

Buckley, S. E., Leverett, M. C., 1942. Mechanism of Fluid Displacement in Sands. Society of Petroleum Engineers.

Burden, R. L., Faires, J. D., 2010. Numerical Analysis Ninth Edition. Cengage Learning.

Chen, Q. Y., Mifflin, R. T., Wan, J., Yang, Y., 2007. A New Multipoint Flux Approximation for Reservoir Simulation. In: SPE Reservoir Simulation Symposium. Houston, USA.

Chen, Y., Durlofsky, L. J., 2006a. Adaptive Local-Global Upscaling for General Flow Scenarios in Heterogeneous Formations. Transport in Porous Media 62 (2), 157–185.

Chen, Y., Durlofsky, L. J., 2006b. Efficient Incorporation of Global Effects in Upscaled Models of Two-Phase Flow and Transport in Heterogeneous Formations. Multiscale Model. Simul. 5 (2), 445–475.

Chen, Y., Durlofsky, L. J., Gerritsen, M., Wen, X. H., 2003. A Coupled Local-Global Upscaling Approach for Simulating Flow in Highly Heterogeneous Formations. Advances in Water Resources 26 (10), 1041–1060.

Chen, Y., Li, Y., 2009. Local-Global Two-Phase Upscaling of Flow and Transport in Heterogeneous Formations. Multiscale Model. Simul. 8 (1), 125–153.

Chen, Y., Li, Y., Efendiev, Y., 2013. Time-of-flight (TOF)-based Two-phase Upscaling for Subsurface Flow and Transport. Advances in Water Resources 54, 119–132.

Christie, M. A., 2001. Flow in Porous Media - Scale Up of Multiphase Flow. Current Opinion in Colloid & Interface Science 6 (3), 236–241.

Christie, M. A., Blunt, M. J., 2001. Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques. SPE Reservoir Simulation Symposium 4 (4).

Deutsch, C. V., 1989. Calculating Effective Absolute Permeability in Sandstone/Shale Sequences. SPE Formation Evaluation 4 (3), 343–348.

Durlofsky, L. J., 1991. Numerical Calculation of Equivalent Grid Block Permeability Tensors for Heterogeneous Porous Media. Water Resources Research 27, 699–708.

Durlofsky, L. J., 1998. Coarse Scale Models of Two Phase Flow in Heterogeneous Reservoirs: Volume Averaged Equations and Their Relationship to Existing Upscaling Techniques. Computational Geosciences 2 (2), 73–92.

Durlofsky, L. J., 2005. Upscaling and Gridding of Fine Scale Geological Models for Flow Simulation. In: 8th International Forum on Reservoir Simulation. Iles Borromees, Stresa, Italy.

Durlofsky, L. J., Jones, R. C., Milliken, W. J., 1997. A Nonuniform Coarsening Approach for the Scale-up of Displacement Processes in Heterogeneous Porous Media. Advances in Water Resources 20 (5-6), 335–347.

Efendiev, Y., 2000. Exact Upscaling of Transport in Porous Media and its Applications. IMA Preprint Series 1724.

Ekrann, S., Aasen, J. O., 2000. Steady-State Upscaling. Transport in Porous Media 41 (03), 245–262.

Gupta, A. D., King, M. J., 1995. A Semianalytic Approach to Tracer Flow Modeling in Heterogeneous Permeable Media. Advances in Water Resources 18, 9–24.

Hilden, S. T., Berg, C. F., 2016. An Analysis of Unsteady Flooding Processes: Varying Force Balance and the Applicability of Steady-State Upscaling. Transport in Porous Media 115 (01), 125–152.

Holden, L., Nielsen, B. F., 2000. Global Upscaling of Permeability in Heterogeneous Reservoirs; The Output Least Squares (OLS) Method. Transport in Porous Media 40 (2), 115–143.

Jacks, H., Smith, O. J. E., Mattax, C. C., 1973. The Modelling of a Three-Dimensional Reservoir with a Two-Dimensional Reservoir Simulator - The Use of Dynamic Pseudo Functions. SPE Journal 13 (03), 175–185.

Lie, K. A., 2016. An Introduction to Reservoir Simulation Using MATLAB: User Guide for the Matlab Reservoir Simulation Toolbox (MRST). SINTEF ICT.

Lie, K.-A., Moyner, O., Krogstad, S., 2015. Application of Flow Diagnostics and Multiscale Methods for Reservoir Management. In: SPE Reservoir Simulation Symposium. Texas, USA.

MathWorks, 2019. Piecewise Cubic Hermite Interpolating Polynomial. `https://se.mathworks.com/help/matlab/ref/pchip.html`, Accessed: 2019-05-04.

Matringe, S., Gerritsen, M., 2004. On Accurate Tracing of Streamlines. In: SPE Annual Technical Conference and Exhibition. Houston, USA.

Møyner, O., Krogstad, S., Lie, K.-A., 2015. The Application of Flow Diagnostics for Reservoir Management. SPE Journal 20 (2), 306–323.

Napov, A., Notay, Y., 2012. An Algebraic Multigrid Method with Guaranteed Convergence Rate. SIAM J. Sci. Comput 34, A1079–A1109.

Natvig, J. R., Lie, K.-A., 2008. Fast Computation of Multiphase Flow in Porous Media by Implicit Discontinous Galerkin Schemes with Optimal Ordering of Elements. Journal of Computational Physics 227, 10108–10124.

Natvig, J. R., Lie, K.-A., Eikemo, B., Berre, I., 2007. An Efficient Discontinous Galerkin Method for Advective Transport in Porous Media. Advances in Water Resources 30, 2424–2438.

Nielsen, B. F., Tveito, A., 1998. An Upscaling Method for One-Phase Flow in Heterogeneous Reservoirs; A Weighted Output Least Squares (WOLS) Approach. Computational Geosciences 2 (2), 93–123.

Norsk Regnesentral, 2003. SAIGUP - Impact of geological uncertainties on production forecasting. `https://www.nr.no/saigup`, Accessed: 2019-05-14.

Notay, Y., 2010. An Aggregation-Based Algebraic Multigrid Method. Electronic Transactions on Numerical Analysis 37, 123–146.

Notay, Y., 2012. Aggregation-Based Algebraic Multigrid for Convection-Diffusion Equations. SIAM J. Sci. Comput 34, A2288–A2316.

Notay, Y., 2019. AGMG software and documentation. `http://agmg.eu`, Accessed: 2019-05-14.

Odsæter, L. H., Berg, C. F., Rustad, A. B., 2015. Rate Dependecy in Steady-State Upscaling. Transport in Porous Media 110, 565–589.

Pickup, G., Ringrose, P. S., Sharif, A., 2000. Steady-State Upscaling: From Lamina-Scale to Full-Field Model. Society of Petroleum Engineers Journal 5 (02), 208–217.

Pickup, G. E., Ringrose, P., Jensen, J. L., Sorbie, K., 1994. Permeability Tensors for Sedimentary Structures. Mathematical Geology 26 (2), 227–250.

Saez, A. E., Otero, C. J., Rusinek, I., 1989. The Effective Homogeneous Behaviour of Heterogeneous Porous Media. Transport in Porous Media 3 (3), 213–238.

Shahvali, M., Mallison, B., Wei, K., Gross, H., 2015. An Alternative to Streamlines for Flow Diagnostics on Structured and Unstructured Grids. SPE Journal 17 (3), 768–778.

Stephen, K. D., Pickup, G. E., Sorbie, K. S., 2001. The Local Analysis of Changing Force Balances in Immiscible Incompressible Two-Phase Flow. Transport in Porous Media 45 (01), 63–88.

Thiele, M. R., 2001. Streamline Simulation. In: 6th International Forum on Reservoir Simulation. Schloss Fuschl, Austria.

Wallstrom, T. C., Christie, M. A., Durlofsky, L. J., Sharp, D. H., 2002a. Effective Flux Boundary Conditions for Upscaling Porous Media Equations. Transport in Porous Media 46, 139–153.

Wallstrom, T. C., Hou, S., Christie, M. A., Durlofsky, L. J., Sharp, D. H., Zou, Q., 2002b. Application of Effective Flux Boundary Conditions to Two-Phase Upscaling in Porous Media. Transport in Porous Media 46 (2-3), 155–178.

Wen, X.-H., Chen, Y., Durlofsky, L. J., 2006. Efficient 3D Implementation of Local-Global Upscaling for Reservoir Simulation. SPE Journal 11 (4), 443–453.

Wen, X. H., Durlofsky, L. J., Lee, S. H., Edwards, M. G., 2000. Full Tensor Upscaling of Geologically Complex Reservoir Descriptions. In: SPE Annual Technical Conference and Exhibition. Dallas, Texas.

Wen, X.-H., Hernandez, J. J. G., 1996. Upscaling Hydraulic Conductivities in Heterogeneous Media: An Overview. Journal of Hyrdology 183 (1-2), 9–32.

Wu, X. H., Efendiev, Y., Hou, T. Y., 2002. Analysis of Upscaling Absolute Permeability. Discrete and Continous Dynamical Systems, Series B 2 (2), 185–204.

APPENDIX A

SATURATION DISTRIBUTIONS

Visualizations of the saturation distributions at different PVI for both single layers and the full model are shown in this appendix. The visualizations are used as a measure of the accuracy of the upscaled models compared to the fine-scale model and in comparisons between the upscaling procedures. In all visualizations, the fine-scale solution is averaged over each coarse block and plotted on the coarse grid. The left column in the visualizations is the averaged fine-scale solution, the middle column is the results obtained with a tracer-based model with the tracer function corresponding to the layer/stack, while the right column is the results obtained with a flow-based upscaling. The global single-phase upscaling method was used in the upscaled models.

The color scale of the visualizations are shown in **Fig. A.1**.

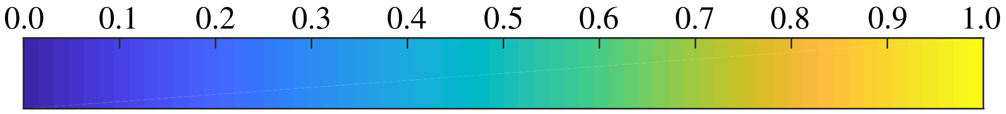
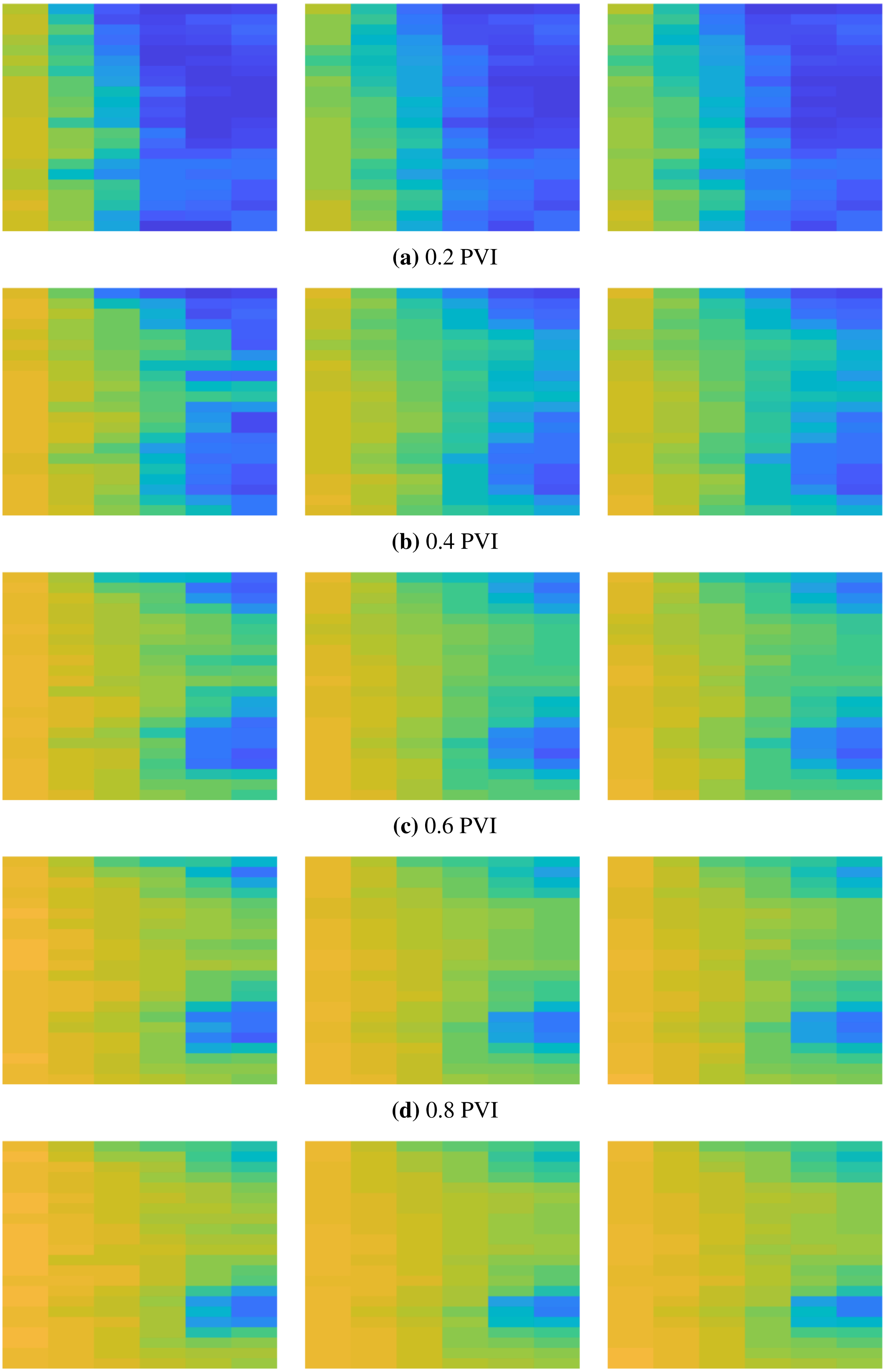


Figure A.1: Colorbar showing normalized scale used in the presented results.

Since the visualizations are hard to tell apart, Fig. A.5 show plots of the relative error between the fine-scale solution and the upscaled solutions and Fig. A.6 shows the relative error between the upscaled solutions.



(a) 0.2 PVI

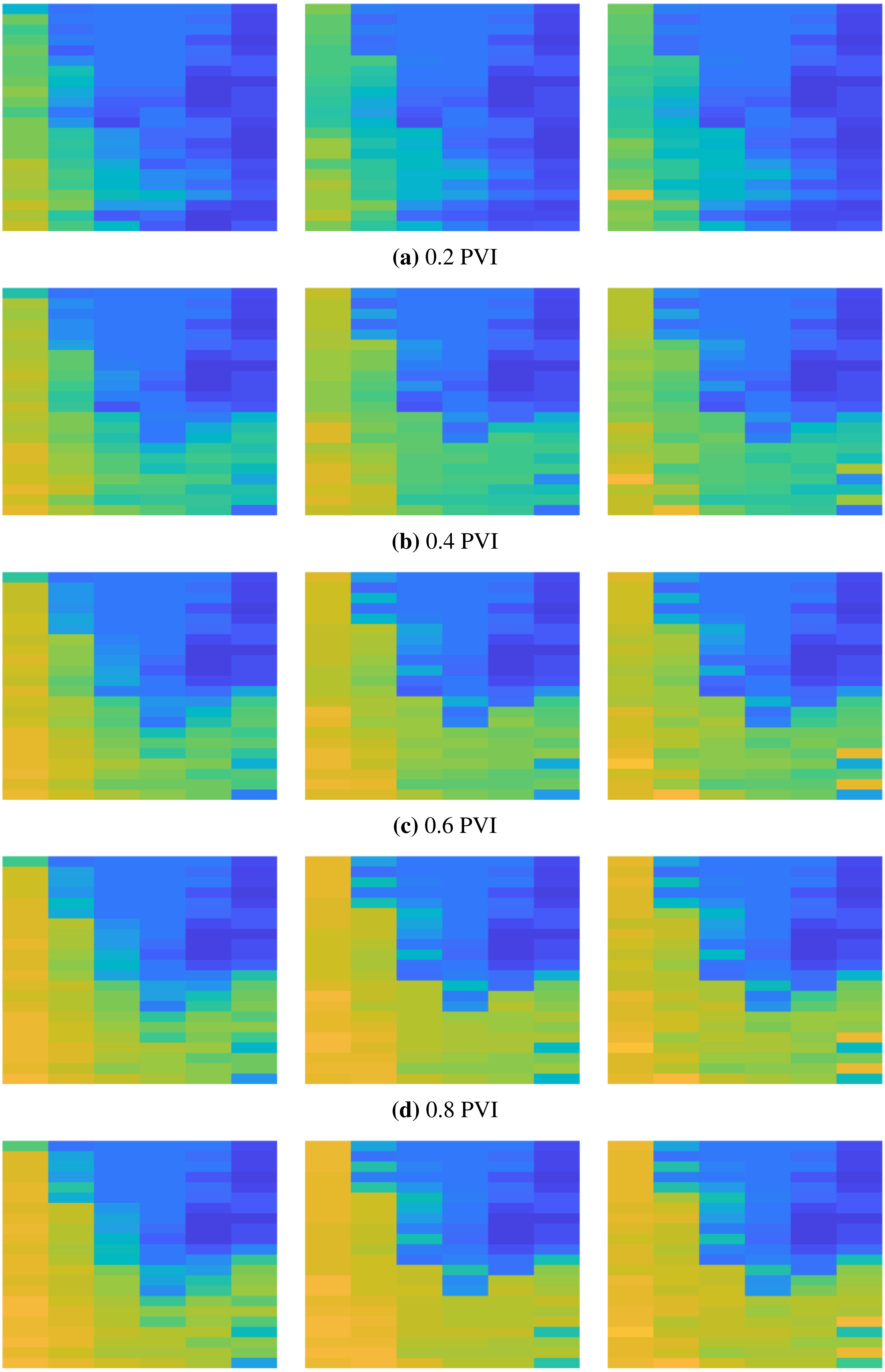
(b) 0.4 PVI

(c) 0.6 PVI

(d) 0.8 PVI

(e) 1.0 PVI

Figure A.2: Saturation distributions at different PVI for layer 26.



(a) 0.2 PVI

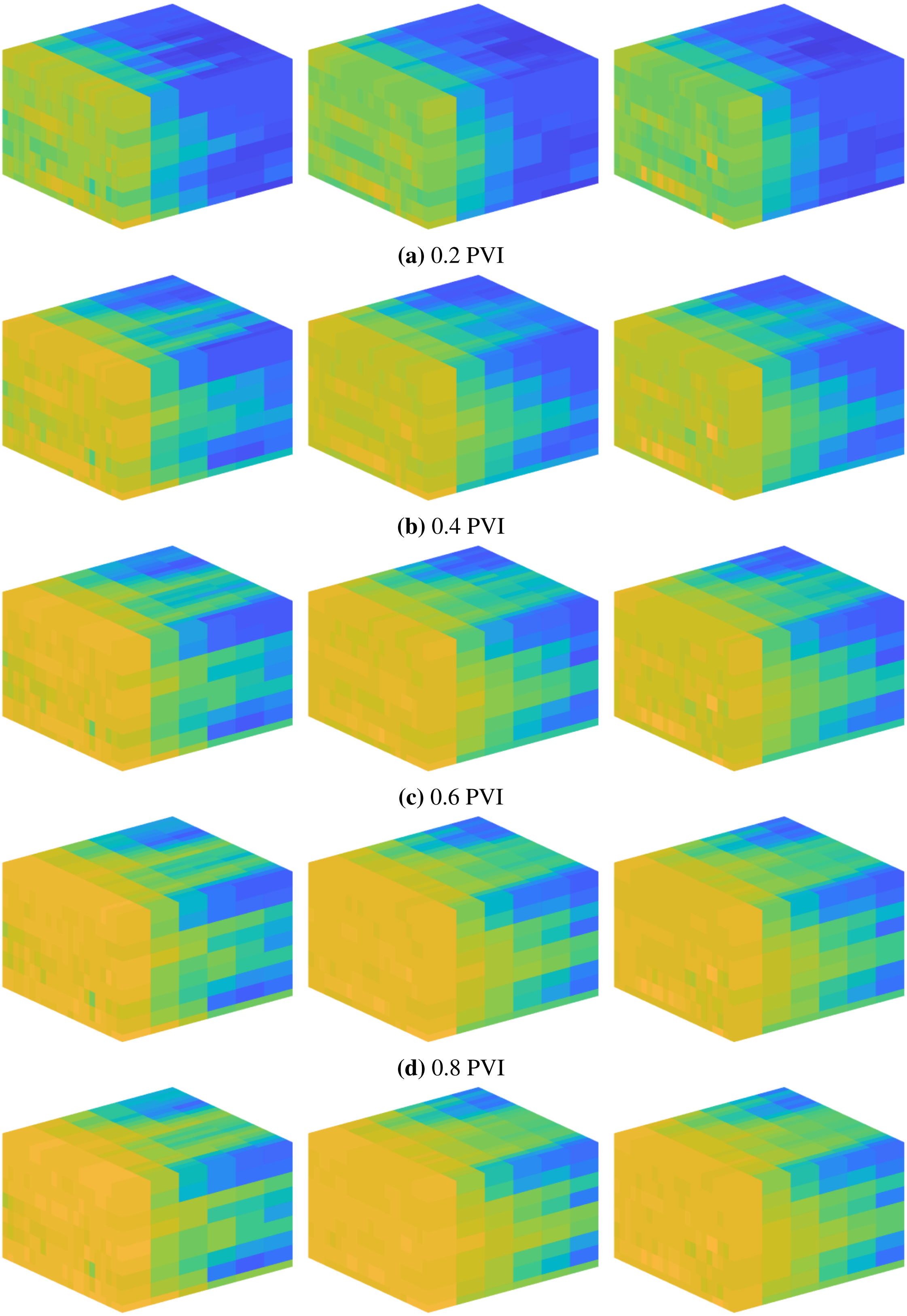
(b) 0.4 PVI

(c) 0.6 PVI

(d) 0.8 PVI

(e) 1.0 PVI

Figure A.3: Saturation distributions at different PVI for layer 44.



(a) 0.2 PVI

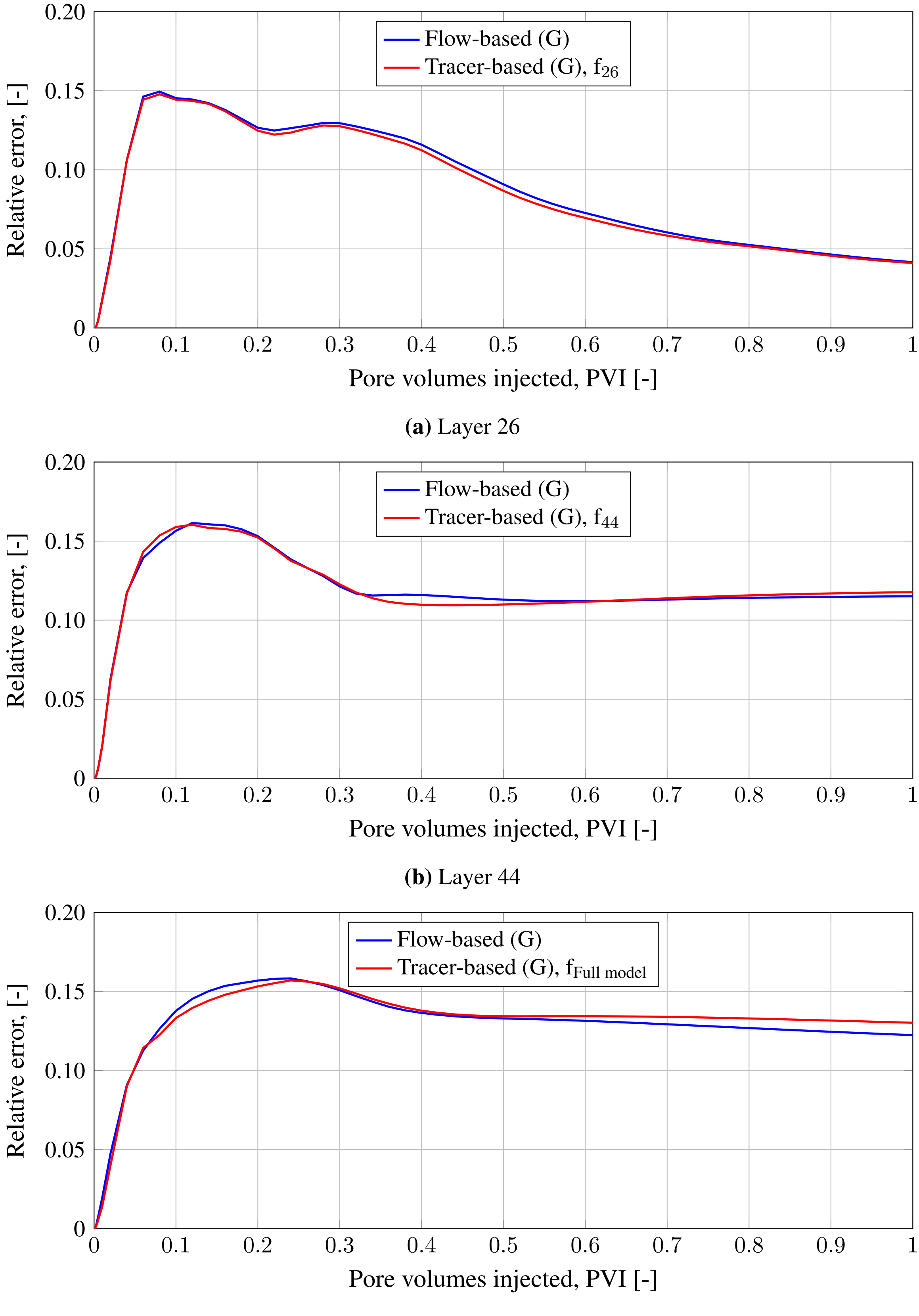
(b) 0.4 PVI

(c) 0.6 PVI

(d) 0.8 PVI

(e) 1.0 PVI

Figure A.4: Saturation distributions at different PVI for the full model.

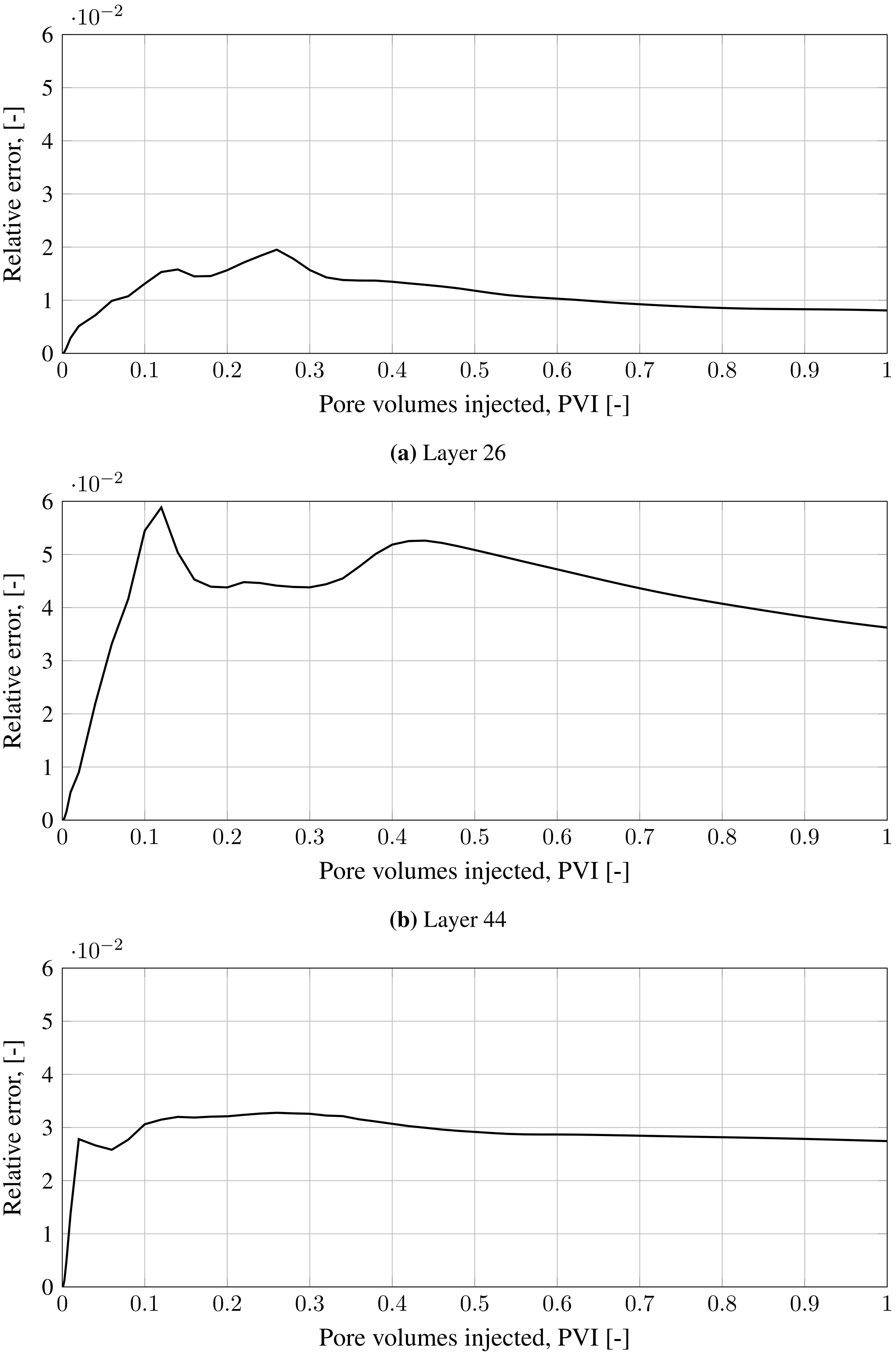


(a) Layer 26

(b) Layer 44

(c) Full model

Figure A.5: Relative error between the fine-scale solution and the upscaled solutions shown in Figs. A.2-A.4.



(a) Layer 26

(b) Layer 44

(c) Full model

Figure A.6: Relative error between the upscaled solutions shown in Figs. A.2-A.4.

APPENDIX B

RELATIVE PERMEABILITY CURVES

The upscaled relative permeability curves obtained by using the tracer functions corresponding to the single layers are shown in this appendix. **Fig. B.1** shows the curves of layer 26, while **Fig. B.2** shows the curves of layer 44.

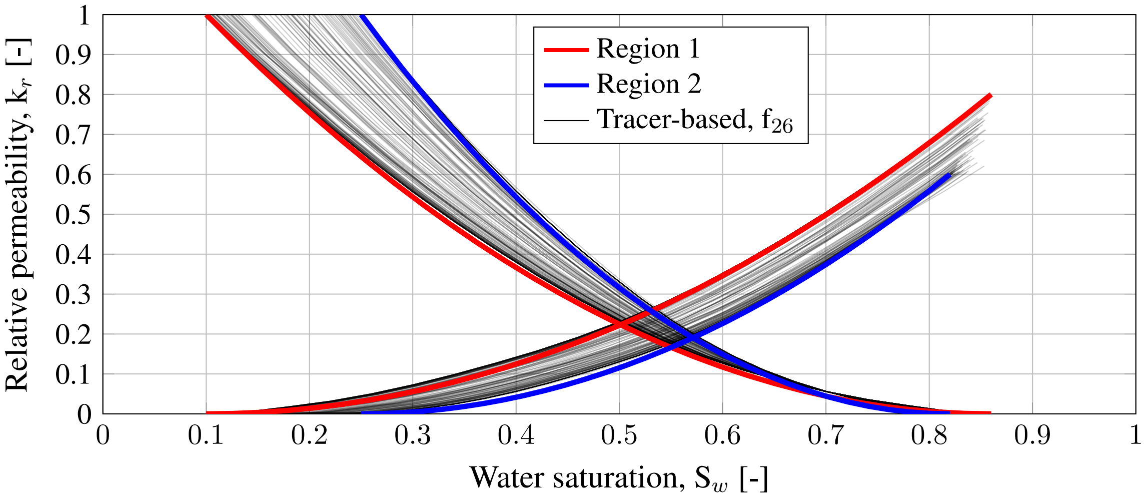


Figure B.1: Tracer-based upscaled relative permeability curves from layer 26.

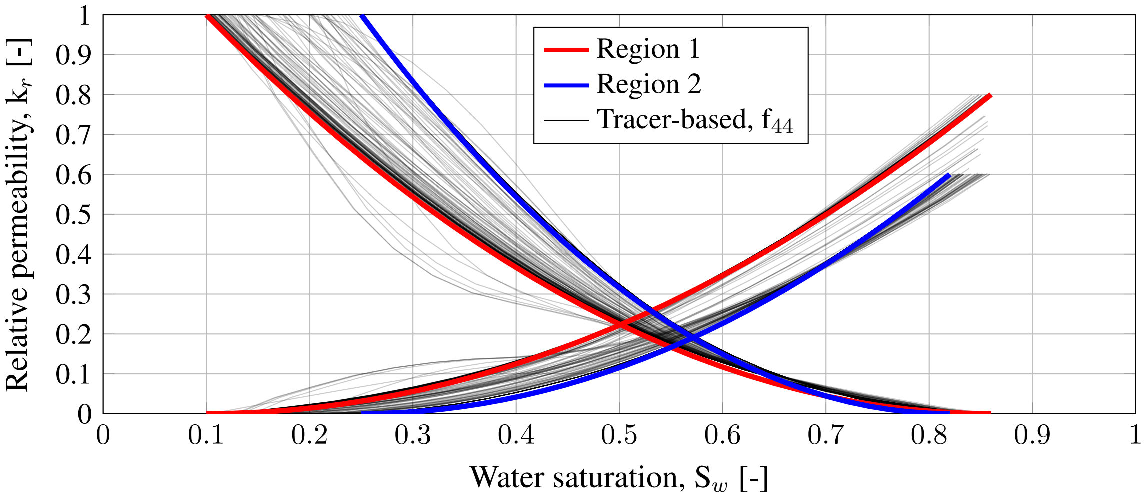


Figure B.2: Tracer-based upscaled relative permeability curves from layer 44.

APPENDIX C

CODE

This appendix includes the *MATLAB*-code required to implement the tracer-based upscaling procedure described in Chapter 3. For an explanation of the data structures and MRST-functions used, see Lie (2016). Since there exist a lot of code for two-phase upscaling in MRST, most of the code shown in this appendix are modifications of the existing code. Some functions have been written from scratch, however, and whether or not the code is a modified version of the standard MRST code is indicated in the caption of the code.

The unmodified two-phase upscaling functions may be found in the steady-state module in MRST. The framework for performing simulations both on both the fine- and coarse-scale may be found in the ad-core module. See the *simulatorWorkflowExample* in MRST for an example of the workflow used to perform simulations on fine and coarse models.

Code C.1: Main framework for two-phase upscaling. The code will run the upscaling procedure with the correct upscaler class in accordance with the upscaling method used.

```
function up = getUpscaledData(cg, rock, fluid, method)
% Load tracer data
if strcmpi(method, 'tracerFlow')
    load('func.mat', 'func');
    fluid_T = initSimpleADIFluid('mu',  [1, 1, 0]*centi*poise,      ...
                                 'rho', [1, 1, 0]*kilogram/meter^3, ...
                                 'n',   [1, 1, 0]);
else
    func    = [];
    fluid_T = [];
end

% Create upscaler
upscaler = TwoPhaseUpscaler(cg.parent, rock, fluid, ...
    'RelpermMethod',    method,       ...
    'relpermdims',      1,            ...
    'RelpermAbsMethod', 'pressure',   ...
    'func',             func,         ...
    'fluid_T',          fluid_T,      ...
    'nrelperm',         20,           ...
    'partition',        cg.partition, ...
    'verbose',          true);

% Set method in results
up.method = method;

% Upscale
[data, report] = upscaler.upscale();

% Store results
up.data   = data;
up.report = report;
end
```

Code C.2: Function for interpolating the upscaled relative permeability data.

```
function up = interpolateRelPerm(up)
% Loop through all blocks
for i = 1:numel(up.data)
    % Set residual saturations
    swir   = up.data(i).swir;
    sor    = up.data(i).sor;
    % Set dimensions
    dims = up.data(i).relpermdims;
    % Create interpolation range
    s_inter = linspace(swir, 1-sor, 20)';

    % Loop through dimensions
    for j = 1:numel(dims)
        % Set current dimension
        d = dims(j);

        % Sort rows of relative permeabilities and saturations to
        % make sure the saturation is monotonicaly increasing
        up.data(i).krW{d} = sortrows(up.data(i).krW{d});
        up.data(i).krO{d} = sortrows(up.data(i).krO{d});

        % Set current data
        s   = up.data(i).krW{d}(:,1);
        krw = up.data(i).krW{d}(:,2);
        kro = up.data(i).krO{d}(:,2);

        % Interpolate
        inter = @(s,k) interp1(s,k,s_inter,'pchip','extrap');
        krw   = inter(s,krw);
        kro   = inter(s,kro);

        % Store new relative permeability data
        up.data(i).krW{d} = [s_inter krw];
        up.data(i).krO{d} = [s_inter kro];
    end
end

end
```

Code C.3: Function for creating the tracer functions. If the tracer function of multiple layers are wanted a for-loop is added.

```
function [func, fluid_T] = createTracerFunction(cg, rock, schedule, fluid)
% Create tracer fluid object. The fluids have equal properties, so this
% ammounts to the injection of a tracer.
fluid_T = initSimpleADIFluid('mu',  [1, 1, 0]*centi*poise, ...
                             'rho', [1, 1, 0]*kilogram/meter^3, ...
                             'n',   [1, 1, 0]);

% Find residual saturations
swir = fluid.swir(fluid.satnum);
sor  = fluid.sor(fluid.satnum);

% Initial states
p_res    = 300*barsa;
state0   = initResSol(G, p_res, [swir 1-swir]);
state0_T = initResSol(cg.parent, p_res, [0 1]);

% Simulation models
model   = TwoPhaseOilWaterModel(cg.parent, rock, fluid);
model_T = TwoPhaseOilWaterModel(cg.parent, rock, fluid_T);

% Run simulations
gravity reset on
[¬, states]   = simulateScheduleAD(state0, model, schedule);
[¬, states_T] = simulateScheduleAD(state0_T, model_T, schedule);

% Create cell arrays with saturation data
s  = cellfun(@(s) (s.s(:,1)-swir)./(1-swir-sor), states, ...
    'UniformOutput', false);
sT = cellfun(@(s) s.s(:,1), states_T, 'UniformOutput', false);

% Bin tracer data
[N, edges, bin] = histcounts(cell2mat(sT),100);
for i = 1:numel(edges)-1
    x(i,1) = 0.5*(edges(i)+edges(i+1));
end

% Find average saturation in each bin
y = accumarray(bin, cell2mat(s))./N';

% Generate fit function
func = fit(x,y,'linearinterp');
end
```

Code C.4: Creates an upscaled fluid structure which may be used in coarse-scale simulations.

```
function f = getUpscaledFluid(cg, fprop, up)
% Define one region for each coarse grid block
prop.satnum = 1:cg.cells.num;

% Set upscaled residual saturations in each grid block
prop.swir = [up.data.swir];
prop.sor  = [up.data.sor];

% Add relative permeabilities to properties structure
for i = 1:numel(up.data)
    prop.krW{i} = up.data(i).krW{1};
    prop.krO{i} = [flipud(1-up.data(i).krO{1}(:,1)) ...
        flipud(up.data(i).krO{1}(:,2))];
end

% Copy remaining fields (e.g. density and viscosity) from the fluid
% used on the fine-scale
fns = fieldnames(fprop);
for i = 1:numel(fns)
    if ¬isfield(prop, fns{i}) % Check existence
        prop.(fns{i}) = fprop.(fns{i});
    end
end

% Create upscaled fluid
f = initADIFluidOW(prop);

end
```

Code C.5: Modified version of the class definition for the two-phase upscaler. The added fields are *func* and *fluid_T* which enables the use of tracer simulations in the *upRelPerm*-function.

```
classdef TwoPhaseUpscaler < OnePhaseUpscaler
    %Two phase upscaling

properties
    RelpermMethod    % Relperm upscaling method
    RelpermAbsMethod % Abs-perm upscaling used in relperm upscaling
    nrelperm         % Number of point for relperm curve
    pcow             % Capillary pressure
    npcow            % Number of capillary pressures
    pcowgrav         % Whether to include gravity in pcOW upscaling
    func             % Tracer function
    fluid_T          % Tracer fluid
    values           % Saturation values
    relpermdims      % Dimensions to upscale relperm in
    savesat          % Save saturation distributions

end
```

Code C.6: Modified options for the two-phase upscaler class. The options now include the tracer function and fluid.

```
function upscaler = TwoPhaseUpscaler(G, rock, fluid, varargin)
upscaler = upscaler@OnePhaseUpscaler(G, rock, 'fluid', fluid, ...
    varargin{:});

upscaler.RelpermMethod    = [];
upscaler.RelpermAbsMethod = 'pressure';
upscaler.nrelperm         = 20;
upscaler.pcow             = true;
upscaler.npcow            = 100;
upscaler.pcowgrav         = true;
upscaler.relpermdims      = upscaler.dims;
upscaler.savesat          = false;
upscaler.func             = [];
upscaler.fluid_T          = [];
upscaler.values           = [];

upscaler = merge_options(upscaler, varargin{:});
end
```

Code C.7: Modified version of the *upscaleBlock* function in the two-phase upscaler. An option for tracer-based upscaling has been added.

```
elseif strcmpi(upscaler.RelpermMethod, 'tracerFlow')
    up = @() upRelPerm(block, data, upscaler.RelpermMethod, ...
        'func',      upscaler.func,              ...
        'dp',        upscaler.dp,                ...
        'values',    upscaler.values,            ...
        'absmethod', upscaler.RelpermAbsMethod, ...
        'fluid_T',   upscaler.fluid_T,           ...
        'dims',      upscaler.relpermdims,       ...
        'savesat',   upscaler.savesat,           ...
        'nsat',      upscaler.nrelperm,          ...
        'verbose',   true);
```

Code C.8: Modified version of the function header for *upRelPerm*. An option for the tracer function and fluid has been added.

```
function [updata, report] = upRelPerm(block, updata, method, varargin)
% Upscaling of relative permeability
opt = struct(...
    'nsat',        20,         ... % Number of upscaled sat. values
    'fluid_T',     [],         ... % Tracer fluid
    'values',      [],         ... % Specify the values
    'func',        [],         ... % Tracer function
    'dims',        1:3,        ... % What dimensions to upscale
    'dp',          1*barsa,    ... % Pressure drop
    'savesat',     false,      ... % Save saturation distributions
    'absmethod',   'pressure', ... % One-phase upscaling method
    'verbose',     false       ... % Print progress to console
    );
opt = merge_options(opt, varargin{:});
```

Code C.9: Modified version of the helper function *getValues* in *upRelPerm*. An option for tracer-based upscaling has been added.

```
function values = getValues(block, updata, method, nvals)

switch method
    case {'flow', 'tracerFlow'}
        values = linspace(0, 1, nvals)';

    case {'capillary', 'capillary-viscous-dist'}
        assert(isfield(updata, 'pcOW'), ...
            'Run capillary curve upscaling first');
        swUMin = updata.pcOW(1,1);
        swUMax = updata.pcOW(end,1);
        sW     = linspace(swUMin, swUMax, nvals)';
        values = interp1(updata.pcOW(:,1), updata.pcOW(:,2), sW);

    case 'viscous'
        ffdata = upFracFlowOW(block, []);
        swUMin = ffdata.ffW(1,1);
        swUMax = ffdata.ffW(end,1);
        sW     = linspace(swUMin, swUMax, nvals)';
        values = interp1(ffdata.ffW(:,1), ffdata.ffW(:,2), sW);

    case {'capillary_grav', 'capillary-viscous-dist_grav'}
        assert(isfield(updata, 'pcOW_bot'), ...
            'Run gravity capillary curve upscaling first');
        swUMin = updata.pcOW_bot(1,1);
        swUMax = updata.pcOW_bot(end,1);
        sW     = linspace(swUMin, swUMax, nvals)';
        values = interp1(updata.pcOW_bot(:,1), updata.pcOW_bot(:,2), sW);

    otherwise
        error(['Method ''' method ''' not recognized.']);
end

end
```

Code C.10: Modified version of the helper function *valueDistribution* in *upRelPerm*. An option for tracer-based upscaling has been added.

```
function [sW, sWff] = valueDistribution(block, method, val, savesat)
% Get the saturation distribution for the current value, depending on the
% method chosen. The returned saturation may be updatad depending on the
% direction later.

sWff  = [];
G     = block.G;
fluid = block.fluid;

% Get saturation values
switch method
    case {'flow', 'tracerFlow'}
        assert(all(isfield(fluid, {'swir','sor','satnum'})), ...
            'Fluid structure does not have necessary fields.');

        sW = val;

        % We have different saturation regions. The sW values are
        % mapped from [0 1] to the different [swir 1-sor] intervals.
        swir = fluid.swir;
        sor  = fluid.sor;

        % sW mapped from scalar to vector of length nRegions
        sW = swir + sW.* ( (1-sor) - swir );

        % Map sW values to each cell
        sW = sW(fluid.satnum);
```

Code C.11: Modified version of the helper function *directionDistribution* in *upRelPerm*. An option to perform tracer-simulations was added where fT is the tracer fluid and f is the tracer function.

```
function sW = directionDistribution(block,method,sW,d,dp,verbose,fT,f)
switch method
    case 'tracerFlow'
        G = block.G;
        isPeriodic = block.periodic;

        % Set pressure drop
        if isPeriodic
            assert( isprop(block, 'bcp') );
            bcp = block.bcp;
            bcp.value(:) = 0;
            bcp.value(bcp.tags == d) = dp;
            bc  = [];
        else
            % Cells neighbouring the outer faces
            nc = @(i) sum(block.G.faces.neighbors(block.faces{d}{i},:),2);

            sWf = sW(nc(1));
            bc = addBC([], block.faces{d}{1}, ...
                'pressure', dp, 'sat', [sWf 1-sWf]);

            sWf = sW(nc(2));
            bc = addBC(bc, block.faces{d}{2}, ...
                'pressure',  0, 'sat', [sWf 1-sWf]);

            bcp = [];
        end

        % Solve steady state flow with tracer fluid
        state1 = simulateToSteadyStateADI(G, block.rock, ...
            fT, sW, 'bc', bc, 'bcp', bcp, 'nIterMax', 100);

        % Find actual saturation with tracer function
        sW = f(state1.s(:,1));

        % Un-normalize
        swir = block.fluid.swir(block.fluid.satnum);
        sor = block.fluid.sor(block.fluid.satnum);
        sW = swir + sW.*( (1-sor) - swir );
```

Code C.12: Added endpoint upscaling to the *upRelPerm* function.

```
% Add endpoint upscaling
block = addDeckForUpscaling(block);
updata = upRelPermEPS(block, updata, method, ...
    'dims',      1,       ...
    'dp',        opt.dp,  ...
    'absmethod', opt.absmethod);
```

Stian Falkfjell Rosvoldaune

Tracer-Based Two-Phase Upscaling

NTNU
Norwegian University of
Science and Technology