Upscaling for Two-phase Flows in Porous Media

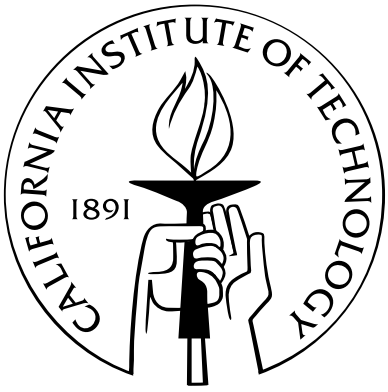
Thesis by

Andrew Westhead

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2005

(Defended January 14, 2005)

Acknowledgements

I am grateful to many of my friends and collegues for the support that they have provided me whilst I have been at Caltech. Firstly, I would like thank my advisor Tom Hou for his patience and guidance, and for giving me the opportunity to develop. There were many times of great frustration during my research but somehow Tom had the foresite and wisdom, and above all faith in me, to know the there would indeed be a light at the end of the tunnel and I would reach it.

I also wish to thanks the many friends that I have made during my time here, especially Chad Schmutzer, Theofilos Strinopoulos, Tobias Kippenberg and Andy Monro. Thanks to Sheila Shull for keeping the department running smoothly.

Thanks to my family for their support.

Thanks to everyone at ExxonMobil, particularly Xiao-Hui Wu and Rossen Pareshkov for their help and advice whilst I was there.

Thanks to Sean Mauch for lending me so many of his books

Finally, I need to thank my true love Kathleen for all the support and keeping me on focus.

Abstract

The understanding and modeling of flow through porous media is an important issue in several branches of engineering. In petroleum engineering, for instance, one wishes to model the "enhanced oil recovery" process, whereby water or steam is injected into an oil saturated porous media in an attempt to displace the oil so that it can be collected. In groundwater contaminant studies the transport of dissolved material, such as toxic metals or radioactive waste, and how it affects drinking water supplies, is of interest.

Numerical simulation of these flow are generally difficult. The principal reason for this is the presence of many different length scales in the physical problem, and resolving all these is computationally expensive. To circumvent these difficulties a class of methods known as upscaling methods has been developed where one attempts to solve only for large scale features of interest and model the effect of the small scale features.

In this thesis, we review some of the previous efforts in upscaling and introduce a new scheme that attempts to overcome some of the existing shortcomings of these methods. In our analysis, we consider the flow problem in two distinct stages: the first is the determination of the velocity field which gives rise to an elliptic partial differential equation (PDE) and the second is a transport problem which gives rise to a hyperbolic PDE.

For the elliptic part, we make use of existing upscaling methods for elliptic equations. In particular, we use the multi-scale finite element method of Hou et al. to solve for the velocity field on a coarse grid, and yet still be able to obtain fine scale information through a special means of interpolation.

The analysis of the hyperbolic part forms the main contribution of this thesis. We first analyze the problem by restricting ourselves to the case where the small scales have a periodic structure. With this assumption, we are able to derive a coupled set of equations for the large scale average and the small scale fluctuations about this average. This is done by means of a special averaging, which is done along the fine scale streamlines. This

coupled set of equations provides better starting point for both the modeling of the large-scale small-scale interactions and the numerical implementation of any scheme. We derive an upscaling scheme from this by tracking only a sub-set of the fluctuations, which are used to approximate the scale interactions. Once this model has been derived, we discuss and present a means to extend it to the case where the fluctuations are more general than periodic.

In the sections that follow we provide the details of the numerical implementation, which is a very significant part of any practical method. Finally, we present numerical results using the new scheme and compare this with both resolved computations and some existing upscaling schemes.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The understanding and modeling of flow through porous media is an important issue in several branches of engineering. In petroleum engineering, for instance, one wishes to model the "enhanced oil recovery" process, whereby water or steam is injected into an oil saturated porous media in an attempt to displace the oil so that it can be collected. In groundwater contaminant studies the transport of dissolved material, such as toxic metals or radioactive waste, and how it affects drinking water supplies, is of interest.

Modeling such flows are difficult and a principal source of the difficultly is the presence of widely different length scales in the problems. In modeling an oil reservoir, for example, geological data will be gathered over an area extending hundreds of meters, if not kilometers. Large scale geological features will be present, such as "faults" or the well pipes, as well as very small scale features such as layers created by sedimentation. For stability when using traditional numerical methods we need to use spatial discretizations that are capable of resolving all these length scales. Even with improving computer technologies this is a formidable task for most data sets, both in terms of memory and computational time. However, for the most part, the resolution required for stability is greater than that needed for engineering purposes. Average properties of the flow, such as the total amount of oil produced, are often of more importance.

To address both these issues, various "upscaling" schemes have been proposed. In an upscaling scheme, one solves only for the average flow features and the effect of the small scale features is modeled. Since capturing average quantities requires less grid resolution, the schemes should use less computer memory and CPU time. The goal of research in this area is to make practical simulation available on limited computer resources. However, the existing upscaling methods often have limitations. Most methods do not have a very

firm mathematical basis, and some rely heavily on experience. Other methods make very restrictive assumption on the nature of the small scale features and are therefore applicable only in a limited range of data sets. A common assumption is that the magnitude of the small scale features is small, an assumption that often does not hold.

The aim of this research has been to develop a framework in which to develop upscaling schemes. We take a particular, simplified model for the porous media flow problem, which is derived in Chapter 2. This model retains the essential difficulty of the problem, namely the presence of multiple scales, and moreover it is one that is used practically. The model consists of an elliptic equation for the fluid pressure and velocity field, and a non-linear hyperbolic transport for the oil-saturation in the porous media. To illustrate the difficulties and motivate the need for upscaling with show some resolved simulations in Chapter 3. At this point we then review some of the previous efforts in designing upscaling schemes and their drawbacks.

The first stage is the formal multiscale analysis of the equations, which is presented in Chapter 4. We perform a multiscale analysis by introducing a small length scale ϵ and small scale variable $y = x/\epsilon$, where x is the large scale variable. To facilitate the initial analysis we assume that there exist only the two distinct length scales outlined above, and that all functions of y are periodic, i.e. the small scale features are periodic. We can apply existing results for the pressure equation to develop a formal equation for the average pressure. The analysis for the hyperbolic transport part are new results and these, along with the numerical results, are the main contribution of this work. We perform a multiscale analysis to derive coupled equations for the average and the fluctuations. In the analysis, we develop a closed set of equations. This is novel since previous results obtain closure only through restrictive assumptions. Closure in our case is obtained by projecting the fluctuations onto a suitable subspace. It turns out that this projection corresponds exactly to averaging along streamlines of the flow. In this subspace, the system becomes closed.

Once these multiscale equations have been derived, we consider the numerical implementation in Chapter 5. We develop a novel approach to sampling the fluctuations in order to compute the small-scale large-scale interaction terms. This is done by using what we call a coarse-grid sub-grid method. We first discretize at the coarse level, using a sufficient number of grid points to resolve average features. Then, within each grid block we solve the equations for the fluctuations at a set number of points. These fluctuations are then

used for computing the interaction terms. The resulting system for the transport part is a coupled hyperbolic system of equations with source terms. In order to solve the pressure equation on such a grid configuration, we make use of a variant of the multiscale finite element method of Hou. It turns out that this complements perfectly the scheme for the transport part.

The numerical method for the hyperbolic part is then tested using a prescribed velocity field and we demonstrate that the average is computed with first-order accuracy. Similarly, we demonstrate the efficacy of MSFEM in capturing the velocity field. We then test the method for the case where the geological data (the permeability) is of the form described above, i.e. with small scale features that are periodic. The results demonstrate that the method captures the average with first order accuracy.

A method for extending the results to the case where permeability is not periodic is then described. We then use this in demonstrating that our method is applicable for practical examples. A sample of some of the results is then given in Chapter 6, and we compare the resulting solutions with those obtained by averaging resolved simulations. Our method captures this average very well. In addition, we compare the computational costs for our method versus resolved computations.

Chapter 2

Modeling Two-Phase Flows

2.1 Overview

In a porous media flow simulation, we are interested in modeling the displacement, within a porous media, of either oil, water or some gas. For the most part of this thesis, including all computations, we will be looking at the case of water-oil simulations. However, for the moment we will stay with the more general case where one of the fluids (but not both) could be a compressible gas. By porous media, we mean a solid with many small voids, or *pores*, potentially connected, through which fluid may flow. The volume fraction of the pores as a total of the whole volume is known as the porosity. Since it is typical to view the pores as a microscale feature, this porosity is a macroscale feature, given pointwise. We usually consider one of the fluids to be displacing the other, as in the case of a oil-water flow where the water is pumped in so as to displace the oil. While the displacing fluid may be immiscible with the fluid being displaced, the displacement does not take place as a piston like process with a sharp interface between the two-fluids. Rather, simultaneous flow of the immiscible fluids takes place within the porous media.

In considering this simultaneous flow we assume, for the present, no mass transfer between the fluids. Mass transfer could potentially occur if there was a chemical reaction taking place between the fluids. Typically, one of the fluids *wets* the porous media more than the other; we refer to this as the wetting phase fluid (and identify it using the subscript w), and we refer to the other as the non-wetting phase fluid (and use the subscript n). Wettability describes the relative preference of a rock (from which the porous media is formed) to be covered by a certain phase. In a water-oil system, water is most often the wetting phase; in a oil-gas system, oil is the wetting phase. We now introduce several concepts

related to multiphase flow, namely saturation and capillary pressure. The saturation, S_k, of a phase k ($k = w, n$) is defined as the fraction of the void volume of the porous medium filled by that phase. Since the two fluids jointly fill the void space, we have

$$S_n + S_w = 1 \ . \tag{2.1}$$

Due to the surface tension and the curvature of the interfaces between the two fluids within the small pores, the pressure in the nonwetting fluid is higher than the pressure in the wetting fluid [28]. The difference between these two pressures is the capillary pressure,

$$p_c = p_n - p_w \ . \tag{2.2}$$

We take as a given fact that the capillary pressure is a unique function of the saturation only [28],

$$p_n - p_w = p_c(S_w) \ . \tag{2.3}$$

Here we have taken as the reference saturation that of the wetting phase, though we could equally have chosen to use that of the non-wetting phase.

In order to model flows in porous media, it is vital to be able to model the velocity field for the flow. It is standard to use Darcy's law as the model for this [28]. For a single phase of fluid in the porous media, Darcy's law relates the fluid velocity $\boldsymbol{v}$ to the pressure p, viscosity μ, density ρ, permeability K, and depth of fluid D via

$$\boldsymbol{v} = -\frac{K}{\mu}(\nabla p - \rho g \nabla D) \tag{2.4}$$

where g is the gravitational constant. We again take this law as an empirical fact [28]. Darcy's law can be thought of a viscous limit of the Navier-Stokes equation, which of course makes sense given how slowly fluid can flow within the porous media. In this equation, we have the permeability which, along with the porosity, is a basic property which characterizes the ease by which fluid can flow in the media. Low permeability characterizes regions where fluid cannot easily penetrate, high permeability where fluid can penetrate.

Darcy's law is extended to multi-phase flow by postulating that the phase pressures p_n and p_w are responsible for the flow within each phase. Thus, equation (2.4) can then be

written for each fluid

$$\boldsymbol{v}_n = -\frac{K_n}{\mu_n}(\nabla p_n - \rho_n g \nabla D) \ , \tag{2.5}$$

$$\boldsymbol{v}_w = -\frac{K_w}{\mu_w}(\nabla p_w - \rho_w g \nabla D) \ . \tag{2.6}$$

Here $\boldsymbol{v}_n$ and $\boldsymbol{v}_w$ are the velocities within the nonwetting and wetting fluids, and the other quantities are also identified for each fluid via the subscripts n or w. K_n and K_w are now the effective permeabilities for the flow for each of the two fluids. Because the simultaneous flow of the two fluids causes each to interfere with the flow of the other, these effective permeability's must be less than or equal to the single-fluid permeability, K, of the medium. Relative permeabilities, which are functions of their respective saturations, are therefore defined by

$$k_{rn} = \frac{K_n}{K} \leq 1 \ , \tag{2.7}$$

$$k_{rw} = \frac{K_w}{K} \leq 1 \ . \tag{2.8}$$

Again we accept as an empirical fact that these relative permeabilities are unique functions of the saturation.

We rewrite Darcy's law now, using the relative permeabilities, to obtain

$$\boldsymbol{v}_n = -\frac{K k_{rn}}{\mu_n}(\nabla p_n - \rho_n g \nabla D) \ , \tag{2.9}$$

$$\boldsymbol{v}_w = -\frac{K k_{rw}}{\mu_w}(\nabla p_w - \rho_w g \nabla D) \ . \tag{2.10}$$

2.2 Two-Phase Flow Equations

If we consider a volume element in the porous media, with porosity ϕ, the usual conservation of mass principle, applied to the single phase case gives

$$-\nabla \cdot (\rho \boldsymbol{v}) + q = \frac{\partial(\phi\rho)}{\partial t} \ . \tag{2.11}$$

Here the q-term represents the external rate of injection of fluid into the volume element. This term will generally be zero, except in those regions in the vicinity of sources or sinks

(which correspond to injection and production wells respectively for our application).

To obtain the corresponding equations for the two-phase flow, by applying the same argument to each fluid phase. Hence, we have

$$-\nabla \cdot (\rho_n \boldsymbol{v}_n) + q_n = \frac{\partial(\phi \rho_n S_n)}{\partial t} , \tag{2.12}$$

$$-\nabla \cdot (\rho_w \boldsymbol{v}_w) + q_w = \frac{\partial(\phi \rho_w S_w)}{\partial t} . \tag{2.13}$$

By combining equations (2.12), (2.13) with (2.9) and (2.10), we obtain the set of simultaneous differential equations that describe two-phase flow

$$\nabla \cdot \left[\frac{\rho_n K k_n}{\mu_n} (\nabla p_n - \rho_n g \nabla D) \right] + q_n = \frac{\partial(\phi \rho_n S_n)}{\partial t} , \tag{2.14}$$

$$\nabla \cdot \left[\frac{\rho_w K k_w}{\mu_w} (\nabla p_w - \rho_w g \nabla D) \right] + q_w = \frac{\partial(\phi \rho_w S_w)}{\partial t} . \tag{2.15}$$

These equations are extremely general in their applicability, as they include the effects of compressibility, capillary pressure, and relative permeability. Furthermore, we also allow for variations in the permeability and porosity.

In flow simulations we are mainly interested in the evolution of the saturation of a particular phase. For instance in the oil-water simulations, we are often interested in the water saturation. The oil saturation can then be immediately obtained via equation (2.1). Using equations (2.14), (2.15) we will now develop a pair of alternative equations. The first of this pair is a "pressure equation" that primarily describes how pressure varies with time and position; the second is a "saturation equation" that describes the variation of saturation with time and position. Not only will these equations be more interesting and insightful, they also are much better suited for numerical solution.

2.2.1 Pressure/Velocity Equation

Our main objective in developing the pressure differential equation is to eliminate the time derivatives of the saturation. To do this, we begin by expanding the time derivatives of

equations (2.12), (2.13) to obtain

$$-\nabla\cdot(\rho_n\boldsymbol{v}_n)+q_n = \left[\rho_n S_n\frac{\partial\phi}{\partial t}+\phi S_n\frac{d\rho_n}{dp_n}\frac{\partial p_n}{\partial t}+\phi\rho_n\frac{\partial S_n}{\partial t}\right], \tag{2.16}$$

$$-\nabla\cdot(\rho_w\boldsymbol{v}_w)+q_w = \left[\rho_w S_w\frac{\partial\phi}{\partial t}+\phi S_w\frac{d\rho_w}{dp_w}\frac{\partial p_w}{\partial t}+\phi\rho_w\frac{\partial S_w}{\partial t}\right]. \tag{2.17}$$

Dividing equation (2.16) by ρ_n and equation (2.17) by ρ_w, and adding the resulting equations and using equation (2.1), we obtain:

$$-\frac{1}{\rho_n}\nabla\cdot(\rho_n\boldsymbol{v}_n)-\frac{1}{\rho_w}\nabla\cdot(\rho_w\boldsymbol{v}_w)+Q_t=\frac{\partial\phi}{\partial t}+\phi S_n c_n\frac{\partial\rho_n}{\partial t}+\phi S_w c_w\frac{\partial\rho_w}{\partial t} \tag{2.18}$$

where we have written

$$Q_t=\frac{q_n}{\rho_n}+\frac{q_w}{\rho_w} \tag{2.19}$$

as the total volume injection rate, and

$$c_n = \frac{1}{\rho_n}\frac{d\rho_n}{dp_n}, \tag{2.20}$$

$$c_w = \frac{1}{\rho_w}\frac{d\rho_w}{dp_w} \tag{2.21}$$

are the phase compressibilities. Note that time derivatives of saturation are now absent from equation (2.18).

Defining an average pressure by

$$p_{avg}=\frac{p_n+p_w}{2} \tag{2.22}$$

the individual phase pressures can then be expressed in terms of the average pressure and the capillary pressure via

$$p_n = p_{avg}+\frac{1}{2}p_c, \tag{2.23}$$

$$p_w = p_{avg}-\frac{1}{2}p_c. \tag{2.24}$$

In addition, let us define *phase mobilities*, λ_n and λ_w, which will be functions of the satu-

ration, by

$$\lambda_n = \frac{k_{rn}}{\mu_n} , \tag{2.25}$$

$$\lambda_w = \frac{k_{rw}}{\mu_w} . \tag{2.26}$$

Then, substitution of equations (2.9) and (2.10) into (2.18) and rearrangement gives the final form of the pressure equation

$$\begin{aligned}
&\left[\frac{1}{\rho_n}\nabla\cdot(\rho_n\lambda_n)+\frac{1}{\rho_w}\nabla\cdot(\rho_w\lambda_w)\right]\nabla p_{avg}+\left[\frac{1}{2\rho_n}\nabla\cdot(\rho_n\lambda_n)-\frac{1}{2\rho_w}\nabla\cdot(\rho_w\lambda_w)\right]\nabla p_c+Q_t \\
&\quad=\left[\frac{d\phi}{dp_{avg}}+\phi(S_nc_n+S_wc_w)\right]\frac{\partial p_{avg}}{\partial t}+\left(\frac{\phi(S_nc_n-S_wc_n)}{2}\right)\frac{\partial p_c}{\partial t} \\
&\qquad+\ g\left[\frac{1}{\rho_n}\nabla\cdot(\rho_n^2\lambda_n)+\frac{1}{\rho_w}\nabla\cdot(\rho_w^2\lambda_w)\right]\nabla D\ .
\end{aligned} \tag{2.27}$$

At first sight this equation seems much more complex than the equations we started with. However, several of the terms vanish upon making some very reasonable assumptions. To understand the nature of equation (2.27), we note first that the capillary pressure p_c is usually quite small relative to the average fluid pressure p_{avg}. The final term, involving depth, may be regarded as a modification to the source term, Q_t. Finally, we can ignore for the moment the variation of ρ_n and ρ_w with position since we will only consider the case where both phases are fluids. Then, equation (2.27) can be simplified to

$$\nabla\cdot[(\lambda_n+\lambda_w)K\nabla p_{avg}]+Q_t\approx\phi c_t\frac{\partial p_{avg}}{\partial t} \tag{2.28}$$

where c_t is a total compressibility defined by:

$$c_t=\frac{1}{\phi}\frac{d\phi}{dp_{avg}}+(S_nc_n+S_wc_w)\ . \tag{2.29}$$

Thus we see that equation (2.28) and therefore equation (2.27) is basically a parabolic equation. However, while the effects of compressibility may not be fully ignored in reservoir calculations, they usually do not dominate. Indeed, any reservoir simulator must be capable of dealing satisfactorily with multiphase flow of incompressible fluids, for which case, $c_t = 0$. This will be the case for the model we will be considering, where the fluids are oil and water. In a oil-gas simulation compressibility effects would likely have a more significant influence.

In short, equation (2.27) should be regarded as being elliptic, or very nearly elliptic, in nature.

For the incompressible case, in which ϕ, ρ_n and ρ_w are constant, equation (2.28) simplifies to

$$-\nabla \cdot (\boldsymbol{v}_n + \boldsymbol{v}_w) + Q_t = 0 \ . \tag{2.30}$$

If we define a total velocity by:

$$\boldsymbol{v}_t = \boldsymbol{v}_n + \boldsymbol{v}_w \tag{2.31}$$

then

$$\nabla \cdot \boldsymbol{v}_t = Q_t \ . \tag{2.32}$$

The simplicity of this equation indicates the fundamental role that total velocity plays in two-phase flow, as opposed to the velocities of each individual phase.

2.2.2 Saturation Equation

In developing an equation for the saturation we may focus on either the wetting or the nonwetting phase. Here we choose, as is standard, the wetting phase. If we have the solution of equation (2.27), then p_w may be obtained from (2.2) and $\boldsymbol{v}_w$ obtained from (2.9) and (2.10). Equation (2.14) which involves $\boldsymbol{v}_w$, could then be used for the saturation equation.

We can, however, derive a more insightful saturation equation that involves the total velocity field defined by equation (2.31). To do this, we first obtain the wetting phase velocity in terms of the total velocity. From equations (2.2), (2.9), (2.10) and (2.26), we have

$$\begin{aligned} \nabla p_c &= \nabla p_n - \nabla p_w & (2.33)\\ \boldsymbol{v}_n &= -\lambda_n(\nabla p_n - \rho_n g \nabla D) \ , & (2.34)\\ \boldsymbol{v}_w &= -\lambda_w(\nabla p_w - \rho_w g \nabla D) \ . & (2.35) \end{aligned}$$

Combining these equations and rearranging gives

$$\lambda_n \lambda_w \nabla p_c = -\lambda_w \boldsymbol{v}_n + \lambda_n \boldsymbol{v}_w + \lambda_n \lambda_w (\rho_n - \rho_w) g \nabla D \ . \tag{2.36}$$

Using (2.31) to eliminate $\boldsymbol{v}_n$, we obtain

$$(\lambda_n + \lambda_w)\boldsymbol{v}_w = \lambda_w \boldsymbol{v}_t + \lambda_n \lambda_w \left[\nabla p_c + (\rho_w - \rho_n) g \nabla D\right] \ . \tag{2.37}$$

Now define the following functions of saturation

$$f_w = \frac{\lambda_w}{\lambda_n + \lambda_w} \ , \tag{2.38}$$

$$h_w = -\frac{\lambda_n \lambda_w}{\lambda_w + \lambda_w} \frac{dp_c}{dS_w} \ . \tag{2.39}$$

Typical curves of f_w versus S_w is shown in Figure 2.1. The negative sign is included in the definition of h_w to keep it positive, since p_c is a decreasing function of S_w. Equation (2.37) then becomes:

$$\boldsymbol{v}_w = f_w \boldsymbol{v}_t - h_w \nabla S_w + \lambda_n f_w (\rho_w - \rho_n) g \nabla D \ . \tag{2.40}$$

and equation (2.12) can be written in the following form

$$\nabla \cdot (\rho_w h_w \nabla S_w) - \nabla \cdot (\rho_w f_w) \left[\boldsymbol{v}_t + \lambda_n (\rho_w - \rho_n) g \nabla D\right] + q_w = \frac{\partial (\phi \rho_w S_w)}{\partial t} \ . \tag{2.41}$$

To investigate the nature of (2.41) we note that the first term (which involves the capillary pressure) strongly suggests that it is basically parabolic in nature, unless capillary effects are insignificant. In that case, the two center terms that involve velocity and gravity becomes more important, but their significance is not so obvious. If we assume incompressibility so that ρ_n and ρ_w are constant, and also constant porosity ϕ, equation (2.41) becomes

$$\nabla \cdot (h_w \nabla S_w) - \nabla \cdot (\rho_w f_w \boldsymbol{v}_t) - \nabla \cdot (G_w \nabla D) + \frac{q_w}{\rho_w} = \phi \frac{\partial S_w}{\partial t} \tag{2.42}$$

where

$$G_w = f_w \lambda_n (\rho_w - \rho_n) g \tag{2.43}$$

is another function of saturation. To understand the nature of the second term of equation (2.42) we expand it, to obtain

$$\nabla \cdot (f_w \boldsymbol{v}_t) = f_w \nabla \cdot \boldsymbol{v}_t + \boldsymbol{v}_t \cdot \nabla f_w \ . \tag{2.44}$$

In turn, we can write

$$\nabla f_w = \frac{df_w}{dS_w} \nabla S_w \ . \tag{2.45}$$

Now, we are primarily concerned with the form of the saturation equation in the regions of the reservoir between the sources and sinks. Therefore, we take $q_w = Q_t = 0$. Equation (2.32) becomes

$$\nabla \cdot \boldsymbol{v}_t = 0 \tag{2.46}$$

and equation (2.42) simplifies to

$$\nabla \cdot (h_w \nabla S_w) - \frac{df_w}{dS_w} \boldsymbol{v}_t \cdot \nabla S_w = \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (G_w \nabla D) \ . \tag{2.47}$$

Equation (2.47) can be regarded as a non-linear variation of the convection-diffusion equation

$$\mathcal{D} \nabla^2 C - \boldsymbol{v} \cdot \nabla C = \phi \frac{\partial C}{\partial t} \tag{2.48}$$

which governs multidimensional miscible displacement. Here $\mathcal{D}$ is diffusivity and C is concentration.

The first term of equation (2.48) is the diffusion term, and when it dominates, (2.48) behaves like the parabolic heat equation. On the other hand, when the diffusion term is small, the center term, i.e. the convection term, dominates and (2.48) approaches the first-order hyperbolic equation:

$$- \boldsymbol{v} \cdot C = \phi \frac{\partial C}{\partial t} \ . \tag{2.49}$$

Referring back to equation (2.47), we can now see that it may be either parabolic or hyperbolic in nature, depending on the importance of the capillary pressure term relative to the convection term. When capillary pressure effects dominate, h_w is large, and (2.47) behaves like a parabolic problem. When capillary pressure effects are small or absent, or more importantly sometimes, when velocities are large, then the convection term dominates, and (2.47) approaches the first-order non-linear hyperbolic equation

$$\frac{df_w}{dS} \boldsymbol{v}_t \cdot \nabla S_w = \phi \frac{\partial S_w}{\partial t} + \nabla \cdot (G_w \nabla D) \ . \tag{2.50}$$

2.3 Simplified Model Problem

To make definite our mathematical formulation we simplify the model with the following assumptions: the porosity ϕ is constant throughout the media; the effects of compressibility can be ignored; capillary effects can be ignored; and gravitational effects can be ignored. In this case, the governing equations (2.28) and (2.50) become

$$\nabla \cdot (\lambda_t(S_w) K \nabla p_{avg}) = 0 \ , \tag{2.51}$$

$$\frac{\partial S_w}{\partial t} + \nabla \cdot (\boldsymbol{v}_t f(S_w)) = 0 \ . \tag{2.52}$$

where $\boldsymbol{v}_t = -\lambda_t(S_w) K \nabla p_{avg}$ and we have written $\lambda_n + \lambda_w = \lambda_t$. Henceforth, we will refer to these equations as the pressure equation and the saturation equation respectively. In addition, we will drop the use of the subscripts in future so that, for example, $\boldsymbol{v}$ and S are understood to be the total velocity and the saturation of the wetting phase respectively. The saturation equation (2.52) was first derived by Buckley and Leverett [6] and is therefore often referred to as the "Buckley-Leverett" equation. In order to complete the description of the model problem we must provide the forms for the functions λ and f in (2.51) and (2.52) and also provide appropriate initial and boundary conditions for the problem. The form of λ and f would in general be determined by experiment. However, for two-phase flow, the phase mobilities given by

$$\lambda_w(S) = S^2 \ , \tag{2.53}$$

$$\lambda_n(S) = m(1-S)^2 \tag{2.54}$$

provide a good model and this model is widely used. Here m is the *mobility ratio*, which is a number between 0 and 1 that indicates the relative ease by which the non-wetting phase can flow. In our case m will generally be less than 1 since oil does not flow as easily as water within the porous medium. From the expressions above and the expression for λ_t and equation (2.39) we have

$$\lambda(S) = S^2 + m(1-S)^2 \ , \tag{2.55}$$

$$f(S) = \frac{S^2}{S^2 + m(1-S)^2} \ . \tag{2.56}$$

The form of $f(S)$ is shown in Figure 2.1 for several different values of m. The special case

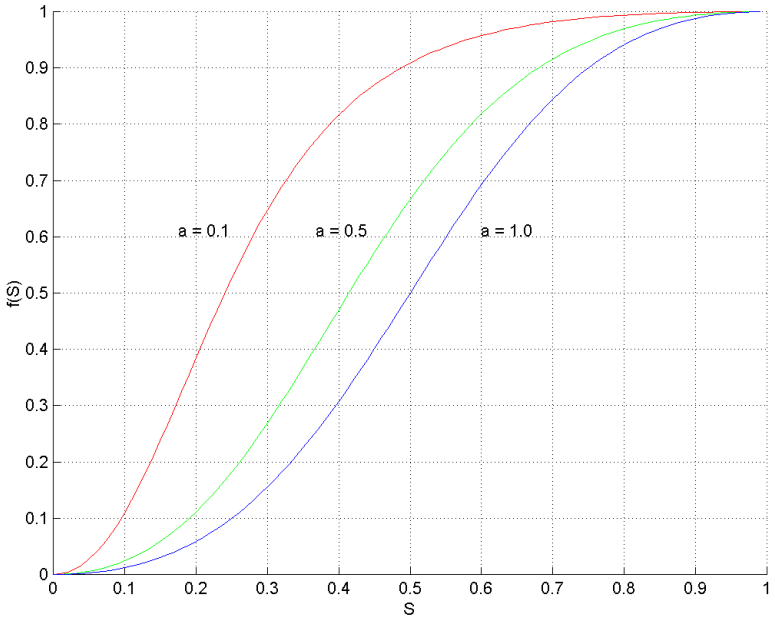


Figure 2.1: Plot of the flux function $f(S)$ for the mobility ratios $m = 0.1, 0.5, 1.0$.

of single phase, or "tracer" flow can be derived more easily, and described in the same framework. The analysis gives us [28]

$$\lambda(S) = 1 , \tag{2.57}$$

$$f(S) = S . \tag{2.58}$$

Note also in the case of one-phase flow that the capillary pressure will be identically zero. One-phase flow is a useful model for the case of solute transport in groundwater flows. In this case contaminated and uncontaminated water, say, will be the two "phases" within the porous media.

We now wish to solve the problem (2.51) and (2.52) for the evolution of the saturation. In general the permeability will be given as input data and this is gathered using some geological survey or seismic imaging. Therefore, we will have to solve the system numerically. We will first describe the solution scheme before any upscaling is done, i.e. our resolved scheme. This will lead us to some general observations as to the nature of the problem and

furthermore, highlight the need for upscaling.

Chapter 3

Resolved Scheme for the Porous Media Flow

To compute the numerical solution of equations (2.51), (2.52) we adopt a solution strategy based on the so-called IMPES (IMplicit Pressure, Explitcit Saturation) scheme [29], [9], which is widely used in reservoir simulation. In this method, a sequential updating of the velocity field and saturation is performed, and each is treated seperately. Initially, the saturation field will be prescribed, along with boundary conditions for the flow field. The first step is to solve for the pressure and velocity field at this initial time. Then, with this velocity field and initial saturation, the saturation is evolved over a small time step with the velocity field kept constant. The resulting saturation is then used to update the pressure and velocity. The process is repeated in this manner up until the time of interest.

The great advantage of this strategy is that the elliptic solver and the hyperbolic solver can essentially be treated seperately and therefore we can take advantage of the existing schemes for each. This will mostly be true in our upscaling scheme as well.

In the next several sections we described the scheme for the resolved calculations. We introduce several important concepts that are common to both our resolved scheme and the upscaling scheme that will be derived. Also, the resolved calculations will demonstrate the need for upscaling more clearly.

3.1 Resolved Scheme for the Pressure Equation

Consider first the pressure equation (2.51), which is elliptic in character. As mentioned, the permeability K will generally be given from geological data and therefore we must solve

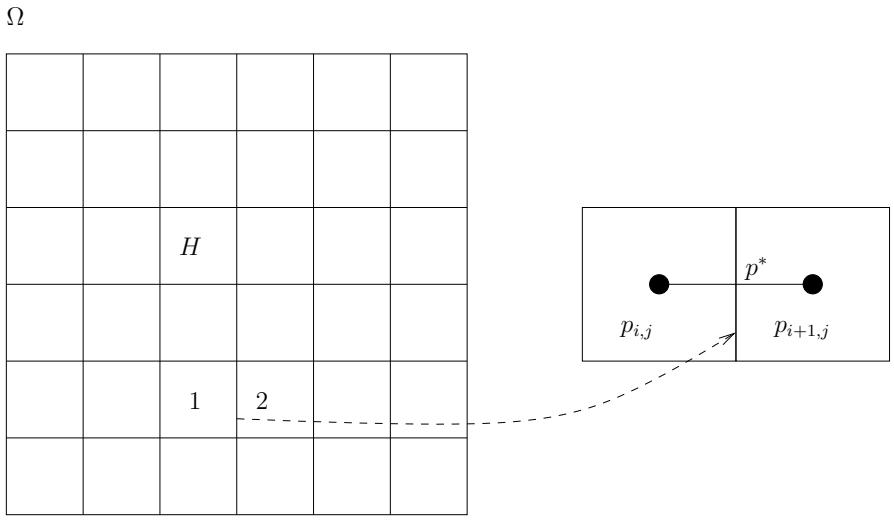


Figure 3.1: Discretization for the resolved pressure equation.

this equation numerically. One of the most important considerations in solving for the velocity field here is that the scheme should preserve the divergence-free property (2.46) in the interior of the domain. Without this property, we will not have conservation of the saturation S which is convected by this velocity field. Although the finite element method (FEM) is often used for solving elliptic equations, in the standard method we cannot force the divergence free condition (2.46) explicitly. While there exist variants that enforce this explicitly, we the method described below is simpler and sufficient for our purposes.

To derive the numerical scheme consider (2.51) on some domain Ω. We divide this domain into rectangular grid blocks, as shown in figure 3.1. First, write $a = \lambda(S)K$ for notational clarity. Then, integrating (2.51) over a grid block H gives

$$\int_H \nabla \cdot (a \nabla p)\, dV = \int_{\partial H} \boldsymbol{n} \cdot (a \nabla p)\, ds \tag{3.1}$$

where $\boldsymbol{n}$ is the outward pointing normal from H. The integral on the right-hand side represents the flux across a cell boundary and this flux across neighboring boundaries should be equal. Assume that the grid is fine enough so that within each grid block H, the saturation S and permeability K can be assumed to be constant. Consider the two adjacent cells 1 and 2 shown in figure 3.1. The pressures at the centers are $p_{i,j}$ and $p_{i+1,j}$ respectively. Similarly, permeabilities are $K_{i,j}$ and $K_{i+1,j}$, and saturations are $S_{i,j}$ and $S_{i+1,j}$. Let the

pressure at the edge be p^*, then the flux continuity can be approximated by

$$h_y a_{i,j} \frac{(p^* - p_{i,j})}{h_x} = h_y a_{i+1,j} \frac{(p_{i+1,j} - p^*)}{h_x} \tag{3.2}$$

if we use a finite difference for the pressure gradient. h_x and h_y denote the grid block sizes in each direction. Rearranging to get an expression for p^* we have

$$p^* = \frac{a_{i+1,j} p_{i+1,j} - a_{i,j} p_{i,j}}{a_{i,j} + a_{i+1,j}} \tag{3.3}$$

Then, the flux across the interface is

$$f_{1,0} = a_{i,j}(p^* - p_{i,j}) \frac{h_y}{h_x} = \frac{a_{i,j} a_{i+1,j}}{a_{i,j} + a_{i+1,j}} (p_{i+1,j} - p_{i,j}) \frac{h_y}{h_x} = a^*_{i,j,1,0} (p_{i+1,j} - p_{i,j}) \frac{h_y}{h_x} \tag{3.4}$$

where we use the notation

$$\frac{a_{i,j} a_{i+k,j+l}}{a_{i,j} + a_{i+k,j+l}} = a^*_{i,j,k,l} \tag{3.5}$$

We can apply the same argument to obtain fluxes over the other edges. Since we have conservation of fluxes over the whole cell we must have

$$\sum f_{k,l} = q_{i,j} \tag{3.6}$$

where $q_{i,j}$ is the external injection rate into the cell (which will be zero except at sources or sinks). Therefore, we have

$$\frac{h_y}{h_x} \sum a^*_{i,j,k,l} (p_{i+k,j+l} - p_{i,j}) = q_{i,j} \tag{3.7}$$

Taking all grid indices i and j in the domain we get a coupled system of linear equations for the pressures at the cell centers. At the edges of the computational domain we usually prescribe no-flow (Neumann) boundary conditions or specify the pressure (Dirichlet) boundary conditions. This system of equations is then solved using a preconditioned conjugate gradient (PCG) scheme or a multigrid solver. Since the permeability K will typically have a large degree of variation it is essential that the solver be able to cope with this. Two particularly robust solvers that we use here are the PCG scheme of Concus *et al.* [10], and the multigrid solver of DeZeeuw [33]. Once we have solved for the pressures, the fluxes

are computed at the center of cell edges using (3.4). Numerical results for the solution are shown in section 3.3.

3.2 Resolved Scheme for the Saturation Equation

We consider now the solution of the saturation equation (2.52), a first-order non-linear hyperbolic problem for the (water) saturation S. In order to understand the structure of the solution we will confine ourselves to the 1- and 2-dimensional cases. We will give a non-rigorous analysis of the problem that provides a useful insight into the structure of the solution.

The full structure of the solution to (2.52) is difficult to analyze in general. However, we can make progress if we note that the mobility ratio m will, in general, not be far from 1 if we are considering oil-water models. This would not be so if we were to be considering the case of an oil-gas model in which the mobility ratio would be extremely small. If the mobility ratio is $O(1)$, then the velocity field is only *weakly* coupled to the saturation, so that changes in the saturation cause only very small corresponding changes in the streamlines. This has been observed by [22] among others, who use it to reduce the number of updates to the velocity field.

Thus, the velocity field $\boldsymbol{v}$ evolves much slower than S and we can consider $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{x})$ for the moment. In section 3.3 we show some calculations that reveal how slowly the streamlines vary as a function of time, which justifies this assumuption. With the velocity field now just a function of the spatial variable $\boldsymbol{x}$ we can write the equation (2.52) as

$$\frac{\partial S(\boldsymbol{x},t)}{\partial t} + f'(S)\left[v_1(\boldsymbol{x})\frac{\partial S(\boldsymbol{x},t)}{\partial x_1} + v_2(\boldsymbol{x})\frac{\partial S(\boldsymbol{x},t)}{\partial x_2}\right] = 0 \tag{3.8}$$

where v_1 and v_2 are the components of $\boldsymbol{v}$. Now introduce the transformations

$$\frac{d\tau}{dx_1} = \frac{1}{v_1(x_1,\eta)} \tag{3.9}$$

$$\frac{d\eta}{dx_1} = \frac{v_2(x_1,\eta)}{v_1(x_1,\eta)} \tag{3.10}$$

then we can rewrite (3.8) in the coordinates $(\boldsymbol{x}, t)$ as

$$\frac{\partial S(\boldsymbol{\chi}, t)}{\partial t} + f'(S)\left[\nu \frac{\partial S(\boldsymbol{\chi}, t)}{\partial \chi_1} + (v_2(\boldsymbol{x}) - \nu v_2(x_1, \eta)) \frac{\partial S(\boldsymbol{\chi}, t)}{\partial \chi_2}\right] = 0 \tag{3.11}$$

in terms of the coordinates $(\boldsymbol{\chi}, t)$ attached to the streamlines $\chi_1 = \tau(x_1; \boldsymbol{x}_o)$ and $\chi_2 = x_2 - \eta(x_1; \boldsymbol{x}_o)$ which pass though the point $\boldsymbol{x}_o$). In the above, $\nu = v_1(\boldsymbol{x})/v_1(x_1, \eta)$. At $\tau = 0$, $\eta = x_{2o}$ for $x_1 = x_{1o}$; and $x_2 = \eta(x_1; \boldsymbol{x}_o)$ is simply the equation of the streamlines of the steady-state velocity field. Along the streamline $\chi_2 = 0$, i.e. $x_2 = \eta$, one obtains $\nu = 1$ and the following equation

$$\frac{\partial S(\boldsymbol{\chi}, t)}{\partial t} + f'(S) \frac{\partial S(\boldsymbol{\chi}, t)}{\partial \tau} = 0 \tag{3.12}$$

which is basically a 1-dimensional equation, except for the additional dependency on $\chi_2 = x_2 - \eta(x_1; x_o)$, which appears only as a parameter identifying the particular streamline. Therefore, in the case of a steady velocity field, we can reduce the 2-dimensional problem to that of a set of 1-dimensional problems along each of the streamlines. This is very useful, as it means that we can determine the structure of the solution by just consideration of the 1-dimensional problem.

The approach outlined above is also the basis of the *streamline method* for solving the porous media flow problem (2.52). In such a method, a set of streamlines $\eta_i = \eta(x_1; \boldsymbol{x}_{o,i})$ is computed and then the method of characteristics is applied upon each of these streamlines to map out the solution. This method also can handle the more practical case of 3-dimensional fields in an exactly similar manner. The purpose of the method is to reduce the effect of numerical diffusion that exists in traditional finite-difference solvers for the flow problem. See [4] and [31] for further details of the method. The drawback with this method is that a very large number of streamlines may be needed for a reasonable solution. Further, in the two-phase flow problem, each time the velocity field is updated, all the streamlines must be updated in addition.

Thus from the simple analysis above, we can see that for the case of modest mobility ratio, with $m = O(1)$ then solution structure for the saturation equation can be understood from the 1-dimensional equation

$$\frac{\partial S}{\partial t} + \frac{\partial f(S)}{\partial x} = 0 \tag{3.13}$$

with $f(S)$ having the same form as before (2.56) We note that the flux function $f(S)$ in our case is nonconvex, though monotonically increasing. Therefore, for general initial data we can expect shocks to form. Before the appearance of shocks we can easily solve this equation by the method of characteristics. In order to see the structure of the shocks that form consider the Riemann problem with initial states $S_l = 1$ and $S_r = 0$. This corresponds to an all water to all oil interface, as shown in figure 3.4 ($t = 0$ plots).

By following the characteristics, we can construct the triple-valued solution shown in figure 3.2. The characteristics velocities are $f'(S)$, so that the profile, seen here at time t,

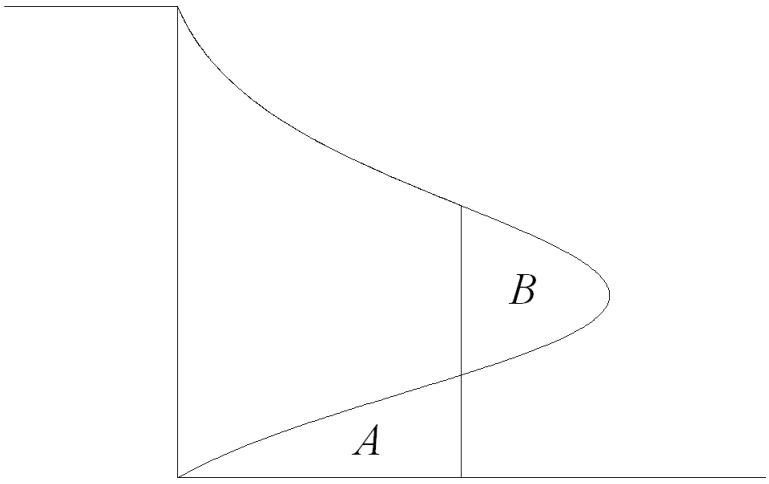


Figure 3.2: Diagram for the construction of the shock via the equal area rule.

is simply the graph of $tf'(S)$ turned sideways. In order to find the size and position of the shock we use the equal area rule (see for example Whitham [32] for a detailed discussion of the construction of the shock), which states that the regions A and B in figure 3.2 must be equal. A simple calculation shows that the size of the shock for this initial data is

$$S_{BL} = \sqrt{\frac{m}{1+m}} \tag{3.14}$$

(the subscript BL denotes that this is for the Buckley-Leverett problem). The speed of the shock for this case can either be determined from the Rankine-Hugoniot condition

$$s = \frac{f(S_{BL}) - f(S_r)}{S_{BL} - S_r} = \frac{f(S_{BL})}{S_{BL}} \tag{3.15}$$

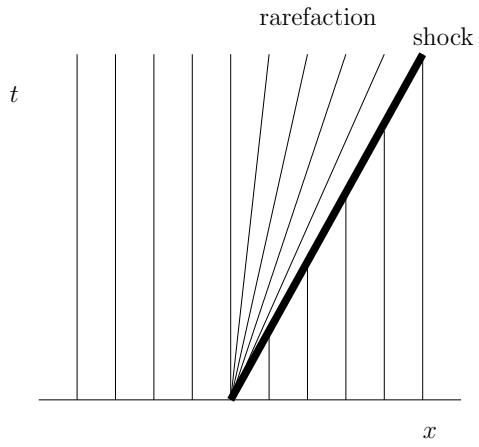


Figure 3.3: Characteristic diagram for the Buckley-Leverett problem.

since $S_r = 0$, or from the equal area construction, which gives the speed as $f'(S_{BL})$. Both the size of the shock and its speed are constant in time. Behind the shock we have a rarefaction wave. The characteristic diagram for the problem is shown in figure 3.3

The resulting weak solutions are shown in figure 3.4 for various values of the mobility ratio m. Note that for smaller values of m the height of the shock is smaller and the speed is greater. The physical interpretation of the solution shown in figure (3.4) is that as water moves in, it displaces a certain fraction S_{BL} of the oil immediately. Once the shock has passed, there is a mixture of oil and water, with less and less oil as time increases. Looking at the output at edge point $x = 1$, one obtains pure oil ($S = 0$) until the shock arrives, followed by a mixture of oil and water (with $S > S_{BL}$) with diminishing returns as time goes on. Note that we never get $S = 1$ at the edge point $x = 1$, and this indicates that it is not possible to force out all the oil in finite time by this means.

Note that the Riemann solution involves both a shock and a rarefraction wave and is called a *compound wave*. If the structure of $f(S)$ was more complicated and there were more inflection points (where $f'(S) = 0$), the solution might involve several shocks and rarefractions.

3.2.1 Solving Nonconvex Riemann Problems

At this point it is worth presenting some theory for the Riemann problem with general nonconvex flux functions $f(S)$. This will be useful not only for the resolved calculations that we are currently considering but also when we derive an upscaled model for the porous

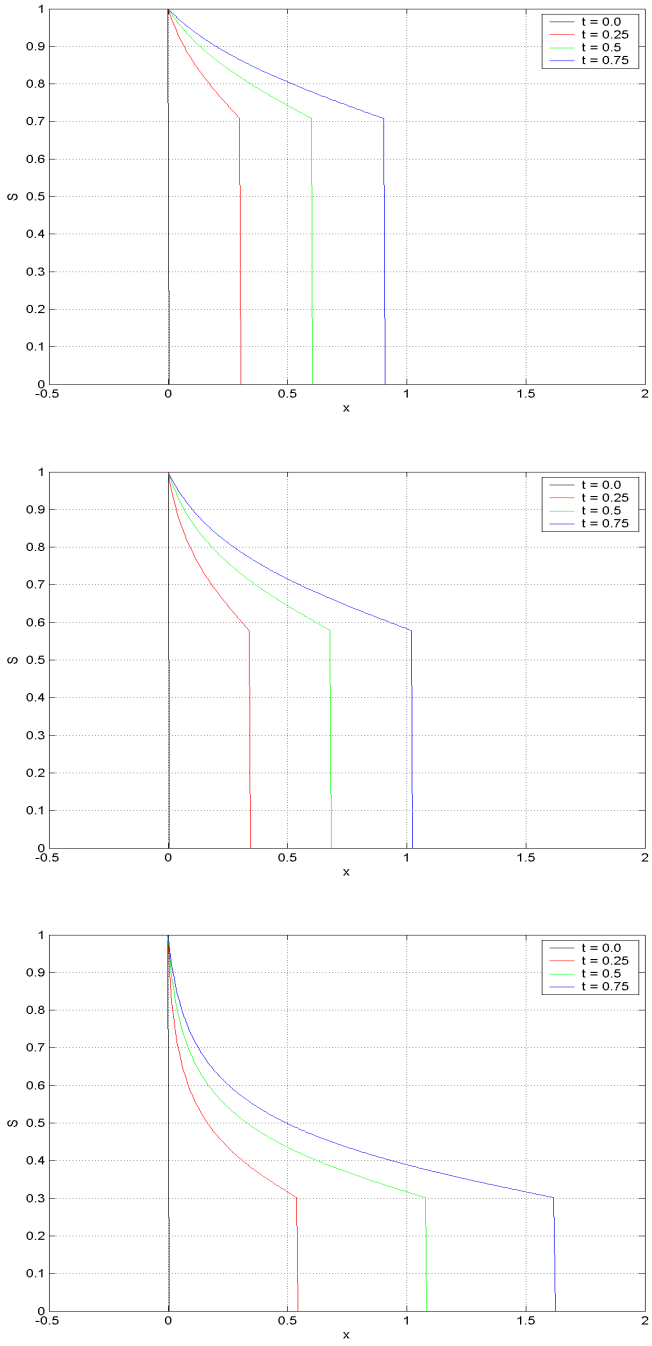


Figure 3.4: 1-dimension saturation profiles at different times for different values of m. Values of m are 1.0, 0.5 and 0.1 respectively.

media flow problem we will again arrive at system whose hyperbolic part has a nonconvex flux function.

To determine the correct weak solution to a non-convex scalar conservation law, we need

an admissibility criterion for shock waves that applies in this case. We have a very general form of the entropy condition, due to Oleinik [26], that applies in the case of nonconvex scalar flux functions $f(S)$:

Theorem 3.1 (Entropy Condition) *A weak solution $S(x,t)$ is the vanishing-viscosity solution to a general scalar conservation law if all the discontinuties have the property that*

$$\frac{f(S) - f(S_l)}{S - S_l} \geq \alpha \geq \frac{f(S) - f(S_r)}{S - S_r} \tag{3.16}$$

for all S between S_l and S_r, where α is the shock speed.

For convex $f(S)$, this requirement reduces to the standard entropy condition.

3.2.2 Convex Hull Construction

Consider a general non-convex flux function $f(S)$ and denote the height of the shock by S^*. The entropy-satisfying solution to a nonconvex Riemann problem can be determined from the graph of $f(S)$ in a simple manner. If $S_r < S_l$, then construct the *convex hull* of the set $\{(S, y) : S_r \leq S \leq S_l \text{ and } y \leq f(S)\}$. The convex hull is the smallest convex set containing the original set. This is shown in figure 3.5 for the case $S_l = 1$, $S_r = 0$.

If we look at the upper boundary of this set, we see that it is composed of a straight line segment from $(0,0)$ to $(S^*, f(S^*))$ and then follows $y = f(S)$ up to $(1,1)$. The point of tangency is precisely the postshock value. The straight line represents a shock jumping from $S = 0$ to $S = S^*$, and the segment where the boundary follows $f(S)$ is the rarefaction wave. This works more generally for any two states (provided $S_l > S_r$) and for any f.

Note that the slope of the line segment is equal to the shock speed α in equation (3.16). The fact that this line is tangent to the curve $f(S)$ at S^* means that $s = f'(S^*)$, the shock moves at the same speed as the as the characteristics at this edge of the rarefaction fan, as seen in figure 3.3. If the shock were connected to some point $\hat{S} < S^*$, then the shock speed $f(\hat{S})/\hat{S}$ would be less than $f'(\hat{S})$, leading to a triple valued solution. On the other hand, if the shock were connected to some point above S^*, then the entropy condition (3.1) would be violated. This explains the tangency requirement, which comes naturally from the convex-hull construction. This same construction works for any $S_r < S_l$ lying in $[0,1]$.

If $S_l < S_r$, then the same idea works, but we instead look at the convex hull of the

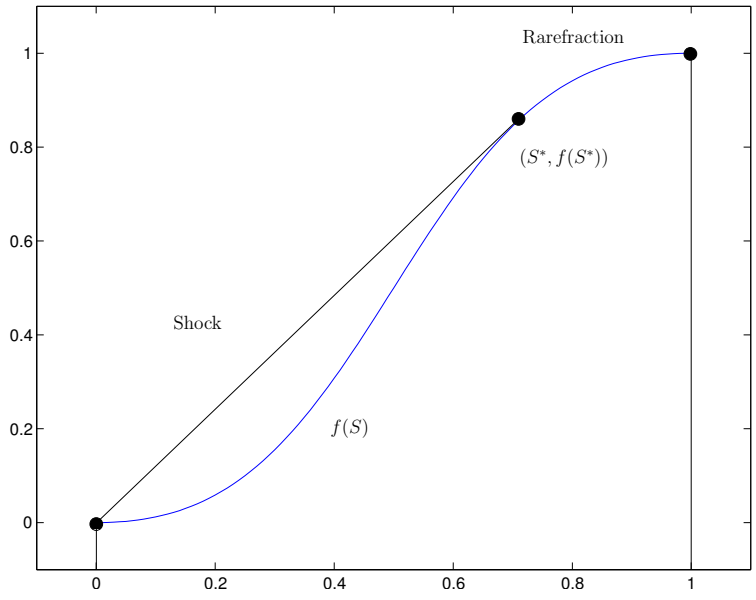


Figure 3.5: Convex-hull construction of the Riemann solution for the Buckley-Leverett equation with $S_l = 1$ and $S_r = 0$.

set of points above the graph $\{(S, y) : S_r \leq S \leq S_l \text{ and } y \geq f(S)\}$. However, we will not consider this case in any greater detail, as we are primarily interested in the case $S_l > S_r$ which corresponds to water displacing oil.

Note that if we were to have convex flux function g, the convex hull construction would give either a single line segment (single shock) if $S_l > S_r$ or the function g itself (single rarefaction) if $S_l < S_r$.

3.2.3 Osher's solution

Osher [27] found a simple representation for the entropy-satisfying similarity solution $S(x,t) = \tilde{S}(x,t)$ for a general nonconvex scalar Riemann problem with arbitrary data S_l and S_r. Let $\xi = x/t$, and set

$$G(\xi) = \begin{cases} \min\limits_{S_l \leq S \leq S_r} (f(S) - \xi S) & \text{if } S_l \leq S_r \\ \max\limits_{S_r \geq S \geq S_l} (f(S) - \xi S) & \text{if } S_r \leq S_l \end{cases} \tag{3.17}$$

Then it can be shown that $\tilde{S}(\xi)$ satisfies the equation

$$f(\tilde{S}(\xi)) - \xi \tilde{S}(\xi) = G(\xi) \tag{3.18}$$

In other words, for any given value of ξ, $\tilde{S}(\xi)$ is the value of S for which $f(S) - \xi S$ is minimized or maximized, depending on whether $S_l \leq S_r$ or $S_r \leq S_l$. We can also write this as

$$\tilde{S}(\xi) = \begin{cases} \underset{S_l \leq S \leq S_r}{\operatorname{argmin}} \left[f(S) - \xi S \right] & \text{if } S_l \leq S_r \\ \underset{S_r \geq S \geq S_l}{\operatorname{argmin}} \left[f(S) - \xi S \right] & \text{if } S_l \leq S_r \end{cases} \tag{3.19}$$

In general the argmin function returns the argument that minimizes the expression and similarly for argmax.

Note that for any fixed S_0 we can replace $f(S) - \xi S$ by $[f(S) - f(S_0)] - \xi(S - S_0)$ in (3.19). Often an appropriate choice of S_0 (e.g. S_l or S_r) makes it is easier to interpret this expression, since it is initimately related to the Rankine-Hugoniot jump condition.

Differentiating the expression (3.18) with respect to ξ gives

$$\left[f'(\tilde{S}(\xi) - \xi \right] \tilde{S}'(\xi) - \tilde{S}(\xi) = G'(\xi) \tag{3.20}$$

Along every ray $x/t = \xi$ in the Riemann solution we have either $\tilde{S}(\xi) = 0$ or else $f'(\tilde{S}(\xi)) = \xi$ (in a rarefaction wave), and hence (3.20) reduces to an expression for $\tilde{S}(\xi)$

$$\tilde{S}(\xi) = -G'(\xi) \tag{3.21}$$

This gives the general solution to the Riemann problem.

The equation (3.18) is particularly useful in the case $\xi = 0$, for which it yields the value $f(\tilde{S}(0)$ along $x/t = 0$. This is the flux value $f(S, (S_l, S_r))$ needed in implementing Godunov's method and generalizations. When $\xi = 0$, (3.18) reduces to

$$f(S, (S_l, S_r)) = f(\tilde{S}(0) = G(0) = \begin{cases} \min\limits_{S_l \leq S \leq S_r} f(S) & \text{if } S_l \leq S_r \\ \max\limits_{S_r \geq S \geq S_l} f(S) & \text{if } S_l \leq S_r \end{cases} \tag{3.22}$$

3.2.4 Finite Volume Scheme for the Saturation Equation

We now have a greater understanding of the structure of the solution to the saturation equation (2.52) in the 1-dimensional case. By the previous analysis, we know that the solution of the fully coupled 2-dimensional problem should have a similar shock and rarefaction structure. We now will discuss the the scheme that we will use to obtain numerical solution

for the 2-D problem.

Since (2.52) is a conservation law in S it is essential that the numerical method we choose respect this property. By using finite volume schemes, which use cell averages of the solution on a numerical grid, we can more easily ensure that the numerical method is also conservative than with other methods. Throughout the following sections we will only consider rectangular grids, as in figure 3.6. As a first step to obtaining a general finite volume

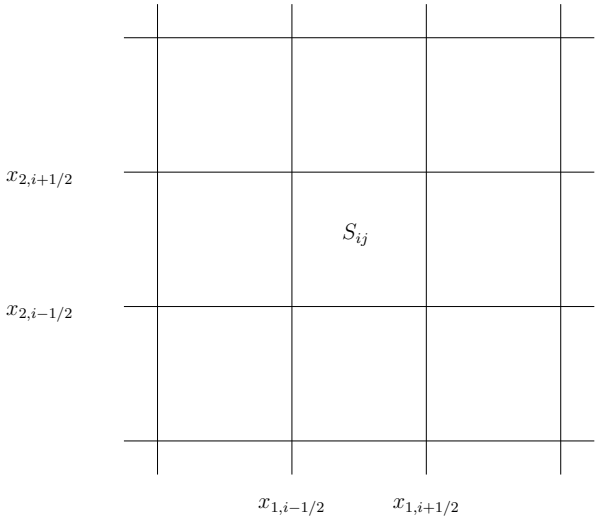


Figure 3.6: Grid for the discretization of the saturation equation.

method, we integrate the saturation equation (2.52) over the cell $\mathcal{C}_{i,j} = [x_{1,i-1/2}, x_{1,i+1/2}] \times [x_{2,i-1/2}, x_{2,i+1/2}]$ to obtain

$$\begin{aligned}\frac{d}{dt}\int\int_{\mathcal{C}_{i,j}} S(x_1,x_2,t) \;=\; & \int_{x_{2,i-1/2}}^{x_{2,i+1/2}} v_1 f(S(x_{1,i+1/2},x_2,t))dx_2 - \int_{x_{2,i-1/2}}^{x_{2,i+1/2}} v_1 f(S(x_{1,i-1/2},x_2,t))dx_2 \\ & + \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} v_2 f(S(x_1,x_{2,i+1/2},t))dx_1 - \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} v_2 f(S(x_1,x_{2,i-1/2},t))dx_1\end{aligned} \tag{3.23}$$

If we now integrate this expression from t_n to t_{n+1} and divide by the cell area $\Delta x \Delta y$, we obtain the fully discrete flux-differencing method of the form

$$S_{i,j}^{n+1} = S_{i,j}^n - \frac{\Delta t}{\Delta x_1}\left[F_{i+1,j}^n - F_{i-1,j}^n\right] - \frac{\Delta t}{\Delta x_2}\left[G_{i,j+1}^n - G_{i,j-1}^n\right] \tag{3.24}$$

where

$$F^n_{i-1/2,j} \approx \frac{1}{\Delta t \Delta x_2} \int_{t_n}^{t_{n+1}} \int_{x_{2,i-1/2}}^{x_{2,i+1/2}} v_1 f(S(x_{1,i-1/2}, x_2, t)) dx_2 dt \tag{3.25}$$

$$G^n_{i,j-1/2} \approx \frac{1}{\Delta t \Delta x_1} \int_{t_n}^{t_{n+1}} \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} v_2 f(S(x_1, x_{2,j-1/2}, t)) dx_1 dt \tag{3.26}$$

and $S^n_{i,j}$ is the numerical approximation to the cell-averages of S

$$S^n_{i,j} \approx \frac{1}{\Delta x_1 \Delta x_2} \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} \int_{x_2-1/2}^{x_2+1/2} S(x_1, x_2, t) dx_1 dx_2 \tag{3.27}$$

The numerical fluxes $F^n_{i-1/2,j}$ and $G^n_{i-1/2,j}$ at each edge are typically computed from the data (numerical approximation) S^n at the previous time. Different methods for evaluating these terms give rise the multitude of different schemes available. In the numerical simulations here, we use the robust wave propagation methods of LeVeque [24] that are implemented in the freely available software package CLAWPACK [23]. Wave-propagation algorithms are based upon solving a Riemann problem at each interface between grid cells and using the resulting wave structure to update the solution in the grid cell on either side. This is, of course, the basis for other methods for conservation laws, going all the way back to Godunov's method.

To see how incorporate the results for the non-convex flux into the scheme, consider using dimension splitting as a means to solve the problem (2.52), i.e. splitting the problem into

$$\frac{\partial S}{\partial t} + v_1 \frac{\partial \left(f(S)\right)}{\partial x_1} = 0 \tag{3.28}$$

$$\frac{\partial S}{\partial t} + v_2 \frac{\partial \left(f(S)\right)}{\partial x_2} = 0 \tag{3.29}$$

In the x-sweeps, we start with the cell averages $S^n_{i,j}$ at time t^n and solve the 1-dimensional problem (3.28) along each row of cells $\mathcal{C}_{i,j}$ with j fixed, updating $S^n_{i,j}$ to an intermediate value $S^*_{i,j}$,

$$S^*_{i,j} = S^n_{i,j} - \frac{\Delta t}{\Delta x_1} \left[F^n_{i+1,j} - F^n_{i-1,j}\right] \tag{3.30}$$

Then, in the y-sweeps, we use the $S^*_{i,j}$ values as data for solving (3.29) along each column

of cells with i fixed, which results in our approximation for $S_{i,j}^{n+1}$,

$$S_{i,j}^{n+1} = S_{i,j}^{*} - \frac{\Delta t}{\Delta x_2}\left[G_{i+1,j}^{*} - G_{i-1,j}^{*}\right] \tag{3.31}$$

An advantage of looking first at this method is that it is easier to see how to enforce the entropy condition (3.16). For example, consider the x-sweeps. From equation (3.28) the numerical fluxes should approximate

$$F_{i+1,j}^{n} \approx \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} v_1 f(S(x_{1,i-1/2}, x_2, t))dt \tag{3.32}$$

$$\approx v_1 \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(S(x_{1,i-1/2}, x_2, t))dt \tag{3.33}$$

(second approximation from the fact that the velocity field is slowly varying). Here $S(x_{1,i-1/2}, x_2, t)$ is taken to be the exact solution of (3.28) with initial data being the numerical solution from the previous time step. Since the initial data consists of the piecewise constant cell averages, we may compute this integral exactly by integrating along the characteristics using Osher's similarity solution for the Riemann problem. This also gives the correct entropy satisfying solution by construction. Since the solution is constant along the characteristics we compute $F_{i+1,j}^{n} = v_1 f(\tilde{S}_{i-1/2,j})$ where $\tilde{S}_{i-1/2,j}$ is the value along $x/t = 0$ in the entropy satisfying solution to the Riemann problem between the states $S_{i-1,j}$ and $S_{i,j}$, which is computed using (3.22). The y-sweeps are then treated in a similar manner.

The above gives a first-order scheme, similar to Godunov's method. Second-order corrections can be introduced in the usual way by incorporating Lax-Wendroff type terms that approximate higher derivatives in the Taylor expansion of the exact solution about (x, t). The modification to $F_{i+1,j}^{n}$ for our case will be of the form

$$\tilde{F}_{i-1/2,j} = \frac{1}{2}\left|s_{i-1/2}\right|\left(1 - \frac{\Delta t}{\Delta x_1}\right)\tilde{\mathcal{W}}_{i-1/2,j} \tag{3.34}$$

where $\tilde{\mathcal{W}}_{i-1/2,j}$ is *limited* version of the wave $\mathcal{W}_{i-1/2,j} = S_{i,j} - S_{i-1,j}$ and $s_{i-1/2,j}$ is the corresponding Rankine-Hugoniot speed,

$$s_{i-1/2,j} = \frac{f(S_{i,j}) - f(S_{i-1,j})}{S_{i,j} - S_{i-1,j}} \tag{3.35}$$

As usual, the limiter limits the influence of this correction so that it is only applied in smooth regions away from the shock. It is also important that the CFL number be less than 1 for stability. In the case of dimension splitting this is satisfied if we have $C \leq 1$ where

$$C = \max \left| v_i \frac{\Delta t}{\Delta x_i} f'(S) \right| \tag{3.36}$$

(no summation in i). Here we maximize over the entire range of S that appears in the solution, which in our case will be $[0, 1]$.

In addition to the dimension splitting scheme, it is possible to derive an *unsplit* scheme for the hyperbolic solver. The details of this method can be found in [24] and so will not be given here. An unsplit scheme is more useful when using a non-uniform mesh and with hyperbolic systems, both of which will be used later.

3.3 Numerical Results for the Resolved Numerical Scheme

To complete the modeling for the flow problem we must specify a domain and flow (or pressure) boundary conditions. In addition we must also specify the initial saturation. We consider two models that are fairly common in the reservoir modeling literature. In both instances will have a domain of rectangular shape ($0 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 1$).

3.3.1 The Core-Plug Model

This model corresponds to a 2-D vertical cross section of an initially oil-saturated "core-plug" (which refers to a sample of the porous media obtained from drilling). For this model it is typical to use a layered permeability field. For the pressure/velocity equation the following configuration is used

- The two ends aligned in the horizontal direction will have zero flux condition, i.e. $\boldsymbol{v} \cdot \boldsymbol{n} = 0$ where $\boldsymbol{n}$ is the outward pointing normal
- The two other sides will have prescribed pressures, $p = 1$ at $x_1 = 0$ and $p = 0$ at $x_1 = 1$.
- The main flow will be in the positive x_1-direction.

The initial saturation, S, will be an established shock

- Initial saturation is independent of x_2
- Initial saturation is a linear function of x_1 between inlet and shock. That is, $S = 1$ at inlet ($x_1 = 0$), $S = S_{BL}$ at shock and $S = 0$ in front of the shock $x_1 = b$. Thus, if we position the shock at position $x_1 = b$, initial saturation is given by

$$S_0(x_1) = 1 - \left(\frac{1 - S_{BL}}{b}\right) x_1, \quad 0 \leq x_1 \leq b \tag{3.37}$$

3.3.2 The Five-Spot Model

The second model is a quarter of a "five-spot" well pattern domain. This model corresponds to an areal view of a field with an injection well at $(0, 0)$ and a production well at $(1, 1)$. In this case, we are viewing the field from above and therefore we should not see layering in the permeability but rather a more uniform hetrogeneous structure:

- The pressure/velocity equation is solved with Neumann condition zero flux, $\boldsymbol{v} \cdot \boldsymbol{n} = 0$, on all four sides
- Injection is at lower corner and production well at upper right. Both wells have constant production, $Q = Q_0$ and $Q = -Q_0$ respectively.

Initial saturation, S, will be an established shock:

- Initial saturation is a linear function of $r = \sqrt{x^2 + y^2}$ the distance from the injection well
- Initial saturation is a linear function of r between injection and shock. That is, $S = 1$ at inlet $S = 1$ at inlet ($r = 0$), $S = S_{BL}$ at shock. Thus, if we position the shock at position $r = b$, initial saturation is given by:

$$S_0 = 1 - \left(\frac{1 - S_{BL}}{b}\right) r, \quad 0 \leq r \leq b \tag{3.38}$$

In both models the geological model is provided by prescribing a spatially varying permeability field (note that we assume a constant porosity for the medium). The permeabilities were generated using the package GSLIB [12], which is used extensively both in industry and within the academic community. The basic model for the permeability uses a scalar log-normal distribution.

To model the effects of hetrogeneous layering which is typically found in geological models, GSLIB will accept, as parameters, the correlation lengths l_1 and l_2 in the horizontal and vertical directions respectively. For the core-plug model we will generally take $l_1/l_2 \geq 1$ and for the five-spot we will use $l_1/l_2 = 1$. Cell-centered permeability fields (constant over the grid blocks) are generated, with 128×128 grid blocks in the vertical and horizontal directions. No additional smoothing of the permeability field is done for finer grids used in the numerical methods, so that finer numerical schemes will still see only a 128×128 underlying permeability field. Examples of the permeability are shown in figures 3.7 and 3.10. Note that in these plots the scale of the color scheme is logarithmic, so that in fact the permeability has an extremely large degree of spatial variation.

There are many other additional parameters that can be given to the GSLIB package to generate permeabilities that better model the different types of porous media. The details of these models are outside the scope of the current investigation.

3.3.3 Numerical Observations

We solve the model problems described above on a 1024×1024 grid. This was determined to be sufficient to resolve the flow details fully. First we describe some of the overall characteristics of the resulting solutions.

Figure 3.7 shows the permeability field with the initial streamlines superimposed for the core-plug model. As one might expect, we see that the flow is channeled through regions of high permeability and it avoids the low permeability regions. Figures 3.8 show the saturation contours at several different times for the core-plug model. We see that saturation develops many hetrogeneous "fingers" as the shock progresses and that behind that shock the saturation has a great deal of spatial variability. By comparing with the permeability field note how the saturation generally follows the fast channels of high permeability.

Figure 3.10 shows the streamlines at $t = 0$ for the five-spot model. In this case we see the flow from the injection well at the lower left corner to the production well at upper right corner. Figures 3.11 show the saturation contours at several different times. Again we see saturation fingers along the fast channels, especially towards the production well.

In addition to the solution variation of the solution $S(x_1, x_2, t)$ with space and time measures of the overall reservoir performance are usually calculated as well. An important characteristic is the so-called *fractional flow* which measures the fraction of oil produced at

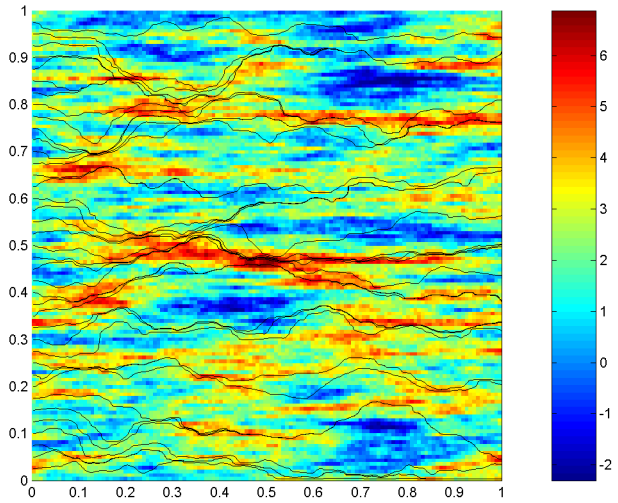


Figure 3.7: Core-plug permeability and streamlines at $t = 0$.

the production with time. This is most easily computed in the case of the core-plug model. In that case, the fractional flow is defined as

$$\Psi(t) = 1 - \frac{\displaystyle\int_0^1 S(1, x_2, t)\boldsymbol{n} \cdot \boldsymbol{v}(1, x_2, t)dx_2}{\displaystyle\int_0^1 \boldsymbol{n} \cdot \boldsymbol{v}(1, x_2, t)dx_2} \tag{3.39}$$

where $\boldsymbol{n} = (1, 0)$ is the outward pointing normal at the edge $x_1 = 1$. Figure 3.12 shows the variation of the fractional flow for core-plog model. At $t = 0$ we have $\Psi = 1$ since initially the domain was oil saturated. After some time however, we see that $\Psi < 1$. This time is the *breakthrough time* corresponding to the first time that water reaches the production well. Accurate determination of breakthough times is also of interest in the performance of the reservoir. Note that instead of time we plot Ψ against pore-volumes injected (PVI). This is a non-dimensional quantity that gives the volume of injected fluid (water in our case) as a fraction of the total pore volume. Since we are assuming constant porosity in our simulations, we have

$$\text{PVI}(t) = \int_0^t \int_0^1 \boldsymbol{n} \cdot \boldsymbol{v}(1, x_2, t)dx_2dt \tag{3.40}$$

3.4 Previous Work on Upscaling and Context of the Present Work

The study of upscaling techniques is by no means new and there have been many contributions to this area. Many of these are relevant to our study here and our discussion will be limited to those closest. Most of the approaches to upscaling are designed to generate some coarse grid description which is approximately equivalent to the underlying fine grid description.

Essentially, the upscaling problem for the whole system can be split into upscaling for the elliptic pressure/velocity equation (which we denote by PVE for short hereafter) and upscaling for the hyperbolic transport equation. For the PVE equation, there have been several upscaling methods developed. Since the permeability data is the principal source of the small scale features, much effort has been devoted to methods for upscaling this quantity. Durlofsky [13], has attempted to find effective permeability properties by dividing the domain into coarse grid blocks, then solving flow problems within each of these. By averaging the resulting flow field within the coarse grid block one can obtain an effective permeability for this grid block. The full PVE is then solved in the domain with the resulting coarse grid effective permeabilities. The limitation of this approach is that boundary conditions must be imposed for the solution of the flow problem within each coarse cell. Since the global flow field is not known apriori, the boundary conditions imposed do not correspond well to the actual boundary conditions, and therefore the resulting effective permeability is dependent on the choice of boundary conditions chosen.

The multi-scale finite element (MSFEM) is a very promising alternative to upscaling the permeability. In this method, coarse grid basis functions are specially constructed which sub-grid features which accurately capture the fine-scale fluctuations. MSFEM has been used successfully to solve the PVE ([19], [20], [17], [16], [8]). Indeed, in the upscaling scheme that we will develop we will be using variant of this method, whereby basis functions are only updated selectively. This leads to a great saving in the amount of computation required.

Less satisfactory progress has been made in developing useful upscaling schemes for the transport equation. The methods that exist can roughly categorized by whether they use a stochastic framework or a deterministic framework. The first approach entails a stochastic formulation of the equations, where the velocity and saturation field are assumed

to have a random component, corresponding to the small scale fluctations. The resulting flow equation for the average saturation then incorporates the expected value and higher statistical moments of these. Langlo and Espedal [22] used this approach to upscale the saturation equation.

Efendiev *et al.* [15], [14], used a hybrid formulation, whereby the upscaled model for the saturation was developed within a fully deterministic framework but the higher order moments of the velocity field were modeled emprically. This was found to be successful in range of cases, though there did exist some serious limitations. The principal difficulty with their scheme was the fact that in developing the model, fluctuations in all quantities were assumed to be small.

The approach in this work is similar to that of Efendiev *et al.* [15]. The main emphasis of this work is to develop an effective scheme for upscaling the saturation equation.

We will initially use a fully deterministic framework to develop an understanding of the effect of the small scale fluctuations upon the average. This will be done at first within the restrictive assumption that all the fine scale fluctuations are periodic. After developing the model in this way, and presenting a numerical implementation to demonstrate it's validity, we will make some minor modifications that will allow us to consider the more general case where the fluctuations are non-periodic.

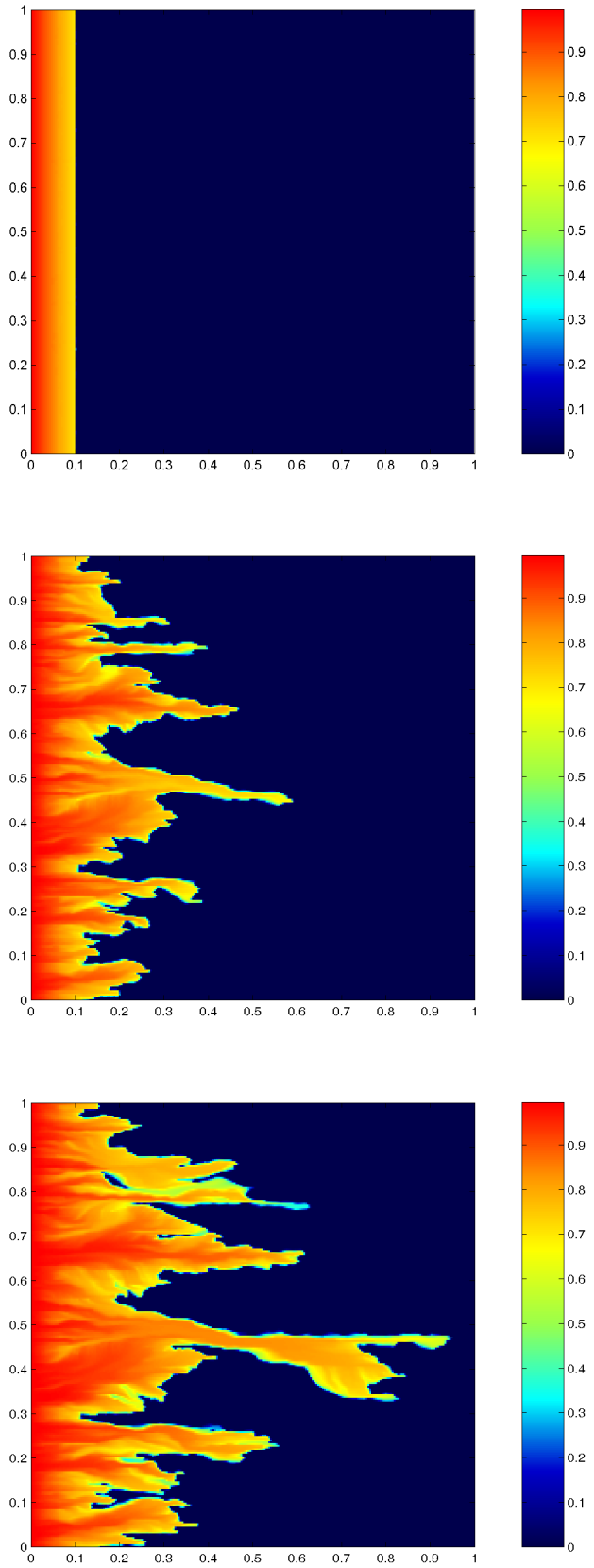


Figure 3.8: Saturation contours for core-plug model, initial data and at intermediate times.

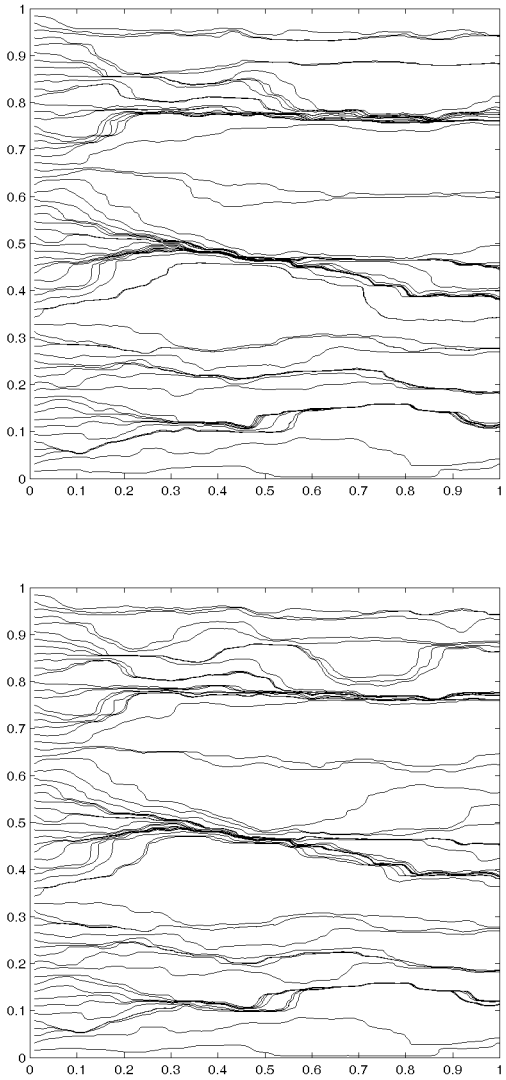


Figure 3.9: Comparison of streamlines at initial time and at the final time, showing that the change in the streamlines is very small.

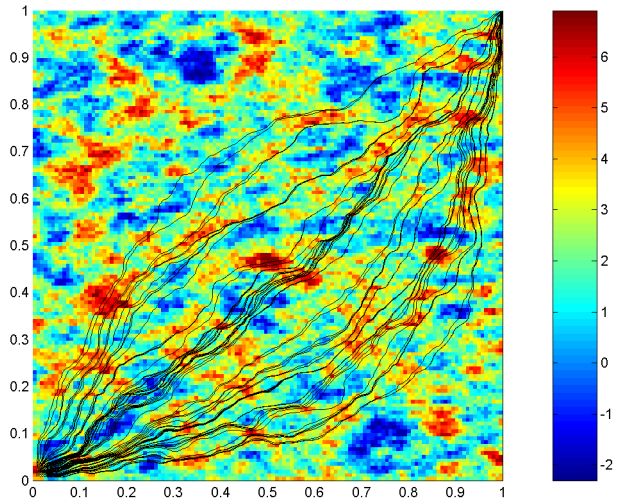


Figure 3.10: Five-spot permeability streamlines at $t = 0$.

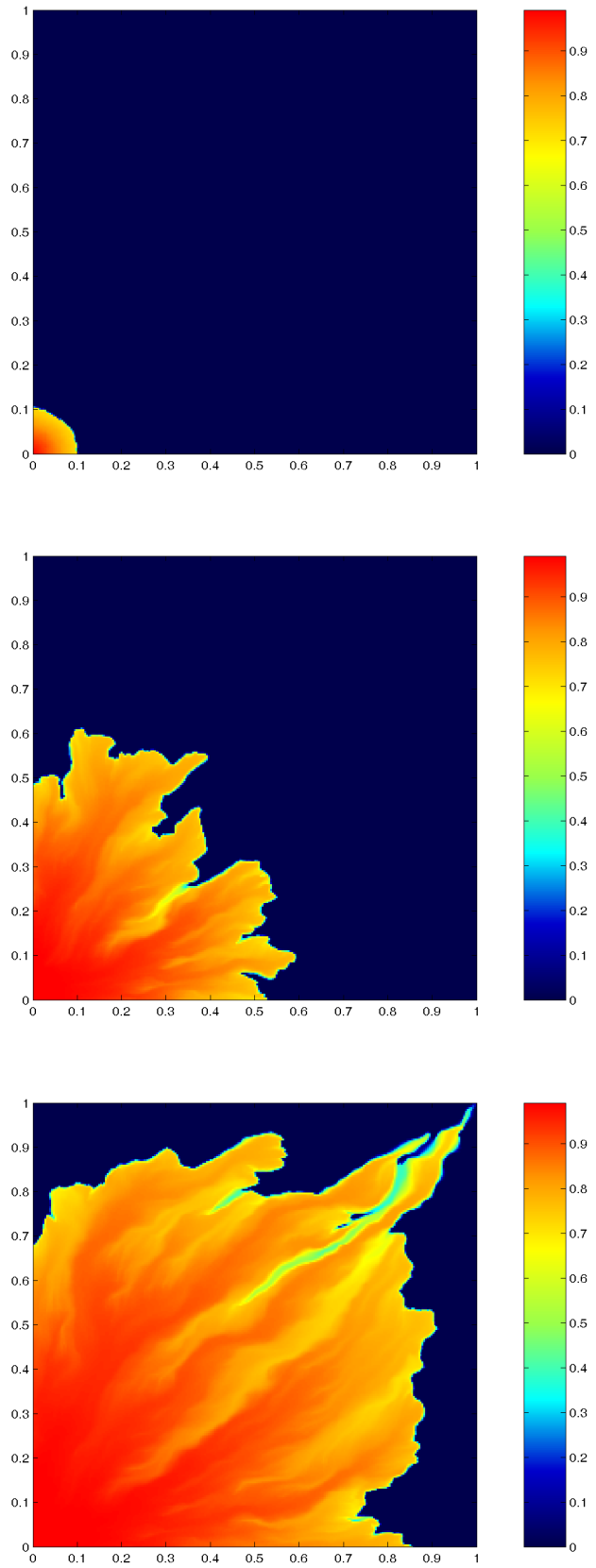


Figure 3.11: Saturation contours for five-spot model, initial data and at intermediate times.

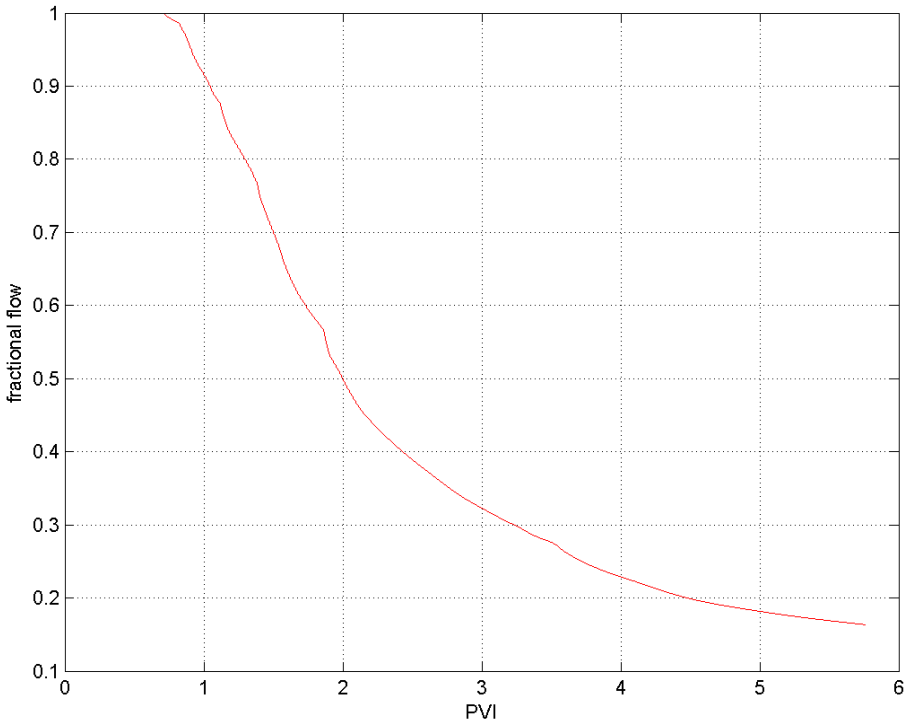


Figure 3.12: Fractional flow against PVI for the core-plug model.

Chapter 4

Multiple Scale Analysis

4.1 Overview

In this chapter we will present the framework for upscaling the porous media flow problem. We perform a multiple-scale analysis for the problem under the assumption that there exists two length scales within the problem: a large scale that captures features at the size of the domain, and a small scale which captures the features within the permeability field. By doing such an analysis, with some assumptions on the nature of the small scale features, we are able to develop equations that model the large scale features and quantify how these are affected by the small scales. From these equations, a numerical scheme for the evolution of the average saturation can be derived easily.

We split up this section in a manner consistent with the overall solution strategy. First, we define some conventions and set up a framework in which to work. Then, we consider the multiple-scale solution for the pressure/velocity equation. This section will mostly be a review of known results. After deriving results for the multiple-scale velocity field, we use this as a starting point for the multiple-scale analysis of the saturation equation. The results for the saturation equation, and their numerical implementation, form the main contribution of this thesis. We develop a coupled set of equations for both the average and the fluctuation. The desirable features of these equations are that they capture all of the important features of the original equation and are *closed*. The closure property is often lacking in previous results, or else is taken care of by making overly restrictive assumptions on the nature of small scales. We achieve this closure by means of a special projection, which we show is equivalent to averaging along streamlines of the flow.

4.2 Formulation of Multiple Scale Model

Consider first the principal source of the small scales in the porous media flow, namely the form of the permeability K. We assume the this has a great deal of spatial variation, and that this is characterized by two scales. The first is a large length scale, on the order of the size of the domain, i.e. $O(1)$ and which we denote by $\boldsymbol{x}$. The second is small length scale, of the order ϵ, with $0 < \epsilon \ll 1$. To model features at this length scale, we introduce the "fast" spatial variable $\boldsymbol{y} = \boldsymbol{x}/\epsilon$. Note that we assume that the two length scales are always distinct. This assumption may not necessarily hold for all types of permeability but it is useful in developing our initial models. With the above length scales defined, the permeability is then given by $K(\boldsymbol{x}, \boldsymbol{y})$. Furthermore, we may write

$$K(\boldsymbol{x}, \boldsymbol{y}) = \overline{K}(\boldsymbol{x}) + K'(\boldsymbol{x}, \boldsymbol{y}) \tag{4.1}$$

where $\overline{K}(\boldsymbol{x})$ represents an "average" of K and $K'(\boldsymbol{x}, \boldsymbol{y})$ is a fluctuation around this average. In general, $\overline{K}(\boldsymbol{x})$ is understood as being the weak limit of K in the limit of $\epsilon \to 0$. As in Chapter 1, we assume that K is a scalar, though in general it could be a tensor.

For our analysis we make the assumption that all functions of the fast variable $\boldsymbol{y}$ are periodic with period Y and that they all lie within the space of square integrable functions. This space will be denoted in the usual way by L^2_Y. For convenience, we will always scale ϵ so that Y is the unit square $[0, 1] \times [0, 1]$. Note that L^2_Y is a Hilbert space if we use the scalar product

$$(u, v)_0 = (u, v)_{L^2_Y} := \int_Y u(\boldsymbol{y}) v(\boldsymbol{y}) d\boldsymbol{y} \tag{4.2}$$

and the corresponding norm

$$\| u \|_0 = \sqrt{(u, u)_0} \, . \tag{4.3}$$

Often in the sections that follow we will drop the use of the subscript 0 when writing this norm. We also introduce the related Sobolev spaces H^m_Y which consists of the set of all functions u in L^2_Y which possess weak derivatives $\partial^\alpha_y u$ in L^2_Y for all $|\alpha| \leq m$. H^m_Y is a Hilbert space with the scalar product

$$(u, v)_m := \sum_{|\alpha| \leq m} (\partial^\alpha u, \partial^\alpha v)_0 \tag{4.4}$$

with the associated norm

$$\| u \|_m = \sqrt{(u,u)_m} = \sqrt{\sum_{|\alpha| \leq m} \| \partial_y^\alpha u \|^2_{L^2_Y}} \ . \tag{4.5}$$

For the most part, we will only be interested in the case of $m = 1$, since our equation (4.32) does not involve higher order derivatives.

Since we will often using the concept of an average quantity, we make this definite by defining, for a function $\phi(\boldsymbol{x}, \boldsymbol{y})$, the average

$$\overline{\phi}(\boldsymbol{x}) = \frac{1}{|Y|} \int_Y \phi(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} \ . \tag{4.6}$$

Note that this is a particular form of the expression for the weak limit of ϕ. The fluctuating part of ϕ will be denoted by ϕ' and is defined in the natural way as

$$\phi'(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x}, \boldsymbol{y}) - \overline{\phi}(\boldsymbol{x}) \ . \tag{4.7}$$

This has clearly zero average, i.e. $\overline{\phi'} = 0$. The average (4.6) can be thought of as a smoothing or spatial "filtering" of the small scales (c.f. Beckie et al [2]).

4.3 Upscaling for the Pressure/Velocity Equation

Consider the elliptic pressure equation (2.52). We make the assumption (which will later be justified) that S consists of an average and a periodic fluctuating part. Then, we have $a = \lambda(S)K$, with $a = \overline{a}(\boldsymbol{x}) + a'(\boldsymbol{x}, \boldsymbol{y})$. Within the framework described above, the form of the solution can be determined using the analysis given in [3]. In equation (2.52) denote the second order elliptic operator by A^ϵ,

$$A^\epsilon = \nabla \cdot (a(\boldsymbol{x}, \boldsymbol{y}) \nabla) \ . \tag{4.8}$$

When differentiating a function $\phi(\boldsymbol{x}, \boldsymbol{y})$, the operator ∇ becomes

$$\nabla_x + \frac{1}{\epsilon} \nabla_y = \frac{\partial}{\partial x_i} + \frac{1}{\epsilon} \frac{\partial}{\partial y_i} \ . \tag{4.9}$$

With this notation, we expand A^ϵ as

$$A^\epsilon = \epsilon^{-2}A_1 + \epsilon^{-1}A_2 + \epsilon^0 A_3 \tag{4.10}$$

where

$$\begin{aligned} A_1 &= \frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial}{\partial y_i}\right) , & (4.11)\\ A_2 &= \frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial}{\partial x_i}\right) + \frac{\partial}{\partial x_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial}{\partial y_i}\right) , & (4.12)\\ A_3 &= \frac{\partial}{\partial x_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial}{\partial x_i}\right) . & (4.13) \end{aligned}$$

We look for an asymptotic expansion of the pressure in the form

$$p_\epsilon = \overline{p}(\boldsymbol{x},\boldsymbol{y}) + \epsilon p_1(\boldsymbol{x},\boldsymbol{y}) + \epsilon^2 p_2(\boldsymbol{x},\boldsymbol{y}) + O(\epsilon^3) \tag{4.14}$$

where each of the functions p_i is periodic in $\boldsymbol{y}$. Substituting the expansion for p_ϵ into our equation $A^\epsilon p_\epsilon = 0$ and gathering together terms with the same powers of ϵ we obtain

$$\begin{aligned} A_1\overline{p} &= 0 , & (4.15)\\ A_1 p_1 + A_2\overline{p} &= 0 , & (4.16)\\ A_1 p_2 + A_2 p_1 + A_3\overline{p} &= 0 . & (4.17) \end{aligned}$$

The first equation

$$\frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial \overline{p}}{\partial y_i}\right) = 0 \tag{4.18}$$

and the fact that $\overline{p}(\boldsymbol{x},\boldsymbol{y})$ is periodic in $\boldsymbol{y}$ implies that $\overline{p} = \overline{p}(\boldsymbol{x})$ by elliptic theory [3]. This result then simplifies the second equation, so that we have

$$\frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial p_1}{\partial y_i}\right) = \left(\frac{\partial a}{\partial y_i}\right)\frac{\partial \overline{p}}{\partial x_i} . \tag{4.19}$$

If we define χ^j as the solution of the following cell problem

$$\frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial \chi^j}{\partial y_i}\right) = \frac{\partial a}{\partial y_j} \tag{4.20}$$

with χ^j periodic in $\boldsymbol{y}$, then the general solution of the second equation is

$$p_1(\boldsymbol{x},\boldsymbol{y}) = -\chi^j \frac{\partial \overline{p}}{\partial x_j} + \tilde{p_1} \ . \tag{4.21}$$

Finally, the solvability condition for p_2 in our third equation

$$\frac{\partial}{\partial y_i}\left(a(\boldsymbol{x},\boldsymbol{y})\frac{\partial p_2}{\partial y_i}\right) = A_2 p_1 + A_3 \overline{p} \tag{4.22}$$

implies that the right-hand side must have mean zero in $\boldsymbol{y}$

$$\int_Y (A_2 p_1 + A_3 \overline{p}) d\boldsymbol{y} = 0 \ . \tag{4.23}$$

This solvability condition gives rise to the homogenized equation for $\overline{p}$,

$$\frac{\partial}{\partial x_i}\left(a^*_{ij}(\boldsymbol{x})\frac{\partial \overline{p}}{\partial x_j}\right) = 0 \tag{4.24}$$

where a^* is a diagonal tensor with

$$a^*_{ij}(\boldsymbol{x}) = \frac{1}{|Y|}\int_Y a(\boldsymbol{x},\boldsymbol{y})\left(1 - \frac{\partial \chi^j}{\partial y_i}\right) d\boldsymbol{y} \tag{4.25}$$

Thus, (4.24), (4.25) and (4.21) define equations for the first two terms in the expansion for the pressure (4.14). Note that the dependence on the fast variable $\boldsymbol{y}$ appears only at $O(\epsilon)$. We can obtain an expression for the velocity field by substituting this pressure expansion into Darcy's law. Doing this we obtain

$$\begin{aligned} \boldsymbol{v}_\epsilon &= -(\overline{a} + a')\left(\frac{\partial}{\partial x_i} + \frac{1}{\epsilon}\frac{\partial}{\partial y_i}\right)(\overline{p}(\boldsymbol{x}) + \epsilon p_1(\boldsymbol{x},\boldsymbol{y}) + \epsilon^2 p_1(\boldsymbol{x},\boldsymbol{y}) + \dots) && (4.26) \\ &= -(\overline{a} + a')(\nabla_x \overline{p} + \nabla_y p_1) + O(\epsilon) \ . && (4.27) \end{aligned}$$

Thus, $\boldsymbol{v}_\epsilon$ has the expansion

$$\boldsymbol{v}_\epsilon = \overline{\boldsymbol{v}} + \boldsymbol{v}' + \epsilon \boldsymbol{v}_1 \tag{4.28}$$

with

$$\overline{\boldsymbol{v}} = \overline{a}\nabla_x\overline{p} + \overline{a'\nabla_y p'} \tag{4.29}$$

$$\boldsymbol{v}' = \overline{a}\nabla_y p' + a'\nabla_x\overline{p} + a'\nabla_y p' - \overline{a'\nabla_y p'} \tag{4.30}$$

The expression for $\boldsymbol{v}_1$ and higher order terms can also be derived. The analysis shows us that if we start with a permeability field with $O(1)$ fluctuations then the resulting velocity field will also have fluctuations which are $O(1)$. As mentioned in the previously, since the mobility λ depends on S, the velocity field is not steady but will change as the S changes throughout the domain.

Note further that the averaged velocity field $\overline{\boldsymbol{v}}$ retains the divergence free property, i.e. $\nabla \cdot \overline{\boldsymbol{v}} = 0$ to $O(\epsilon)$. To see this note that

$$\left(\nabla_x + \frac{1}{\epsilon}\nabla_y\right) \cdot \left(\overline{\boldsymbol{v}} + \boldsymbol{v}'\right) = 0 \ . \tag{4.31}$$

Equating terms with the same power of ϵ, at $O(\epsilon^{-1})$ we get $\nabla_y \cdot \boldsymbol{v}' = 0$. At $O(1)$ we have $\nabla_x \cdot (\overline{\boldsymbol{v}} + \boldsymbol{v}') = 0$. Averaging this equation over Y gives $\nabla_x \cdot \overline{\boldsymbol{v}} = 0$ and hence also $\nabla_x \cdot \boldsymbol{v}' = 0$. Therefore, we see that spatial averaging preserves the divergence-free properties.

4.4 Upscaling for the Saturation Equation

We now consider the problem of homogenization for the hyperbolic saturation equation

$$\frac{\partial S_\epsilon}{\partial t} + \boldsymbol{v}_\epsilon \cdot \nabla f(S_\epsilon) = 0 \tag{4.32}$$

in 2-dimensions. The incompressible velocity field $\boldsymbol{v}_\epsilon$ was shown, in the previous section, to have an $O(1)$ oscillatory component.

In the same way as for the pressure equation, we will confine our analysis the case where the functions of the "fast" variable $\boldsymbol{y} = \boldsymbol{x}/\epsilon$ are periodic. Within this framework we will derive a closed, coupled system of equations for the average $\overline{S}$ and the $O(1)$ fluctuations S'. Closure is obtained making use of a special *streamline* average that eliminates higher-order fluctuations. After developing these expressions we will propose some approximations that allow the methodology to be applied to more general flows for which the oscillations are not

necessarily periodic with respect to the fast variable. In our case, where we are looking at flows more complex than shear flows, we will again see that the nature of the streamlines plays a very important role in determining the effective equation.

We first apply the standard multiple-scale analysis of looking for a formal expansion of the saturation of the form

$$S_\epsilon = \overline{S}(\boldsymbol{x}, t) + S'(\boldsymbol{x}, \boldsymbol{y}, t) + \epsilon S_1(\boldsymbol{x}, \boldsymbol{y}, t, \tau) + O(\epsilon^2) \ . \tag{4.33}$$

Thus, S consists of an average, $\overline{S}$, modified by a fluctuating part S'. We have also introduced in this expansion a possible dependence on a fast time scale $\tau = t/\epsilon$, which appears at the $O(\epsilon)$ level. The justification for such an expansion will be probed further in section 4.4.1. As before, with the expansion for the velocity field, all the terms except $\overline{S}$ have zero mean. The flux function $f(S)$ is expanded in a similar manner

$$f(S_\epsilon) = \overline{f}(\boldsymbol{x}, t) + f'(\boldsymbol{x}, \boldsymbol{y}, t) + \epsilon f_1(\boldsymbol{x}, \boldsymbol{y}, t, \tau) + O(\epsilon^2) \tag{4.34}$$

where again we have that f', $f_1, \ldots$ are periodic in $\boldsymbol{y}$ and f' has with zero mean, i.e. $\overline{f'} = 0$. This expansion is determined solely from the (prescribed) form of f and S_ϵ, with

$$\overline{f} + f' = f(\overline{S} + S') \ , \tag{4.35}$$

$$f_1 = f_S S_1 \ , \tag{4.36}$$

$$f_2 = f_S S_2 + \frac{1}{2} f_{SS} S_1^2 \tag{4.37}$$

where $f_S = \frac{df}{dS}|_{\overline{S}+S'}$, and similarly for the higher-order terms. Note that f_1, f_2 and higher order terms do not necessarily have mean zero.

We again use that fact that for a function $\phi(\boldsymbol{x}, \boldsymbol{x}/\epsilon, t, t/\epsilon)$ we must expand the partial derivatives as

$$\nabla = \nabla_x + \frac{1}{\epsilon} \nabla_y \ , \tag{4.38}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial t} + \frac{1}{\epsilon} \frac{\partial}{\partial \tau} \ . \tag{4.39}$$

Substituting our expansions into the saturation equation and gathering together terms with

the same power of ϵ we obtain the following hierarchy of equations:

$$\epsilon^{-1} \quad : \quad (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_y f' = 0 \tag{4.40}$$

$$\epsilon^{0} \quad : \quad \frac{\partial \overline{S}}{\partial t} + \frac{\partial S'}{\partial t} + \frac{\partial S_1}{\partial \tau} + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_x \overline{f} + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_x f' + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_y f_1 = 0 \tag{4.41}$$

$$\epsilon^{1} \quad : \quad \frac{\partial S_1}{\partial t} + \frac{\partial S_2}{\partial \tau} + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_x f_1 + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_y f_2 = 0 \ . \tag{4.42}$$

To facilitate the analysis, we now introduce some subspaces of L^2_Y and then several Lemmas, which build a framework for multiscale analysis. We introduce the following spaces in L^2_Y:

$$\mathcal{N} \quad = \quad \{u \in H^1_Y : \boldsymbol{v} \cdot \nabla_y u = 0\} \ , \tag{4.43}$$

$$\mathcal{W} \quad = \quad \{\boldsymbol{v} \cdot \nabla_y u : u \in H^1_Y\} \tag{4.44}$$

here $\boldsymbol{v}$ is our velocity field as computed from the pressure equation, so that $\nabla_y \cdot \boldsymbol{v} = 0$. We now also assume that $\boldsymbol{v}$ is bounded and that $v_i \in L^2_Y$. With these spaces, we have the following orthogonal decomposition of L^2_Y:

Lemma 4.1

$$L^2_Y = \mathcal{N} \oplus \overline{\mathcal{W}} \ . \tag{4.45}$$

Proof. In order to prove this lemma we need the following well known theorem:

Theorem 4.1 *Let $\mathcal{H}$ be a Hilbert space and $\mathcal{M} \subset \mathcal{H}$ be a subspace. Then, any element $x \in \mathcal{H}$ has the unique decomposition $x = y + z$, with $y \in \mathcal{M}$, $z \in \mathcal{M}^{\perp}$, where $\mathcal{M}^{\perp}$ denotes the orthogonal complement of $\mathcal{M}$. Furthermore,*

$$\| x - y \| = \min_{\nu \in \mathcal{M}} \| x - \nu \| \tag{4.46}$$

where $\| \cdot \|$ is the associated norm of $\mathcal{H}$.

Thus, to prove Lemma 4.1 we show that $\mathcal{N}$ and $\overline{\mathcal{W}}$ are orthogonal complements in L^2_Y. To do this, first note from their definitions that $\mathcal{N}$ and $\overline{\mathcal{W}}$ are clearly subspaces of L^2_Y (we need to take the closure of $\mathcal{W}$ since this is not a closed space). Now consider u such that $(u, \boldsymbol{v} \cdot \nabla_y w) = 0$ is satisfied for each $w \in H^1_Y$. This implies that $(\boldsymbol{v} \cdot \nabla_y u, w) = 0$ for each $w \in H^1_Y$ and hence $u \in \mathcal{N}$. Since $\boldsymbol{v} \cdot \nabla_y w \in \overline{\mathcal{W}}$ we therefore we have $\overline{\mathcal{W}} \perp \mathcal{N}$ in L^2_Y. Because

L^2_Y is a Hilbert space all the hypotheses of Theorem 4.1 are satisfied and thus Lemma 4.1 follows. ■

In Theorem 4.1, the element $y \in \mathcal{M}$ is called the *orthogonal projection* of x onto $\mathcal{M}$ and the abstract form of the projection is given by (4.46). To derive a more explicit form for the projection in our case, consider using $\mathcal{M} = \mathcal{W}$ in the theorem. Then, from the definition (4.44) for $\mathcal{W}$, for a given $u \in L^2_Y$ the projection $\mathcal{Q} : L^2_Y \mapsto \mathcal{W}$ is the defined as the solution of the minimization problem

$$\| u - \mathcal{Q}(u) \| = \min_{\theta \in H^1_Y} \| u - \boldsymbol{v} \cdot \nabla_y \theta \| \tag{4.47}$$

where $\| \cdot \|$ is the L^2_Y norm defined by (4.5). If we had an orthonormal basis for $\mathcal{W}$ (or its restriction to a finite dimensional subspace) then we could use a least-squares approximation to determine the solution to this problem. In the absence of possessing such a basis we use the following lemma:

Lemma 4.2 *For $u \in H^1_Y$ the projection $\mathcal{Q} : H^1_Y \mapsto \mathcal{W}$ is uniquely given by $\mathcal{Q}(u) = \boldsymbol{v} \cdot \nabla_y \theta$ where $\theta \in H^2_Y$ is the solution of the degenerate elliptic PDE*

$$\nabla_y \cdot (A \nabla_y \theta) = \boldsymbol{v} \cdot \nabla_y u \tag{4.48}$$

with periodic boundary conditions where A is the 2×2 matrix with components $A_{ij} = v_i v_j$.

Proof. First expand the norm in (4.47) via

$$\begin{aligned} \| u - \boldsymbol{v} \cdot \nabla_y \theta \|^2 &= \int_Y (u - \boldsymbol{v} \cdot \nabla_y \theta)^2 \, d\boldsymbol{y} && (4.49) \\ &= \int_Y u^2 - 2u\boldsymbol{v} \cdot \nabla_y \theta + (\boldsymbol{v} \cdot \nabla_y \theta)^2 \, d\boldsymbol{y} && (4.50) \\ &= \int_Y u^2 + 2\theta v_i \frac{\partial u}{\partial y_i} + v_i v_j \frac{\partial \theta}{\partial y_i} \frac{\partial \theta}{\partial y_j} d\boldsymbol{y} && (4.51) \end{aligned}$$

where we have used integration by parts and $\nabla_y \cdot \boldsymbol{v} = 0$ to go from the second equation to the third. Defining

$$\begin{aligned} a(\psi, \phi) &= \int_Y v_i v_j \frac{\partial \psi}{\partial y_i} \frac{\partial \phi}{\partial y_i} d\boldsymbol{y} \, , && (4.52) \\ h(u, \phi) &= -\int_Y \phi v_i \frac{\partial u}{\partial y_i} d\boldsymbol{y} && (4.53) \end{aligned}$$

then the minimization problem (4.47) is equivalent to finding the minimum of

$$J(\theta) := \frac{1}{2}a(\theta,\theta) - h(u,\theta) \tag{4.54}$$

over H^1_Y. It is easy to see that $a(\psi,\phi)$ is a symmetric *semi-positive* bilinear form, i.e. $a(\psi,\psi) \geq 0$ for all $\psi \in H^1_Y$. $a(\psi,\psi)$ is not positive since we can see from its definition that it is zero for $\psi \in \mathcal{N} \subset H^1_Y$. However, for $\phi \in H^1_Y - \mathcal{N}$, $\phi \neq 0$ we have $a(\phi,\phi) > 0$. We first show that $J(\theta)$ attains its minimum over H^1_Y at θ if and only if

$$a(\theta,\phi) = h(u,\phi) \tag{4.55}$$

for all $\phi \in H^1_Y$, and that this minimum is unique up to a function in $\mathcal{N}$. To see this consider $\psi,\ \phi \in H^1_Y$ and $t \in \mathbb{R}$. We have

$$\begin{aligned} J(\theta + t\phi) &= \frac{1}{2}a(\theta + t\phi, \theta + t\phi) - h(u, \theta + t\phi) && (4.56)\\ &= J(\theta) + t\left[a(\theta,\phi) - h(u,\phi)\right] + \frac{1}{2}t^2 a(\phi,\phi)\ . && (4.57) \end{aligned}$$

If $\theta \in H^1_Y$ satisfies (4.55) then (4.57) with $t = 1$ implies

$$\begin{aligned} J(\theta + \phi) &= J(\theta) + \frac{1}{2}a(\phi,\phi) \text{ for all } \phi \in H^1_Y && (4.58)\\ &\geq J(\theta) && (4.59) \end{aligned}$$

with equality only if $\phi = 0$ or $\phi \in \mathcal{N}$. Thus, up to a function in $\mathcal{N}$, θ is a unique minimal point. Conversely, if J has its minimum at θ, then for every $\phi \in H^1_Y$, the derivative of the function $t \mapsto J(\theta + t\phi)$ must vanish at $t = 0$. By (4.57) the derivative is $a(\theta,\phi) - h(u,\phi)$ and so (4.55) follows. Note that A has eigenvalues $\lambda = 0, v_1^2 + v_2^2$ and therefore (4.48) is a degenerate elliptic equation.

Integration by parts of (4.55) and using the fact that $\nabla_y \cdot \boldsymbol{v} = 0$ gives

$$\int_Y \phi \frac{\partial}{\partial y_i}\left(v_i v_j \frac{\partial \theta}{\partial y_j}\right) d\boldsymbol{y} = \int_Y \phi v_i \frac{\partial u}{\partial y_i} d\boldsymbol{y} \tag{4.60}$$

for all $\phi \in H^1_Y$, from which equation (4.48) follows. Returning to (4.47) we see that $\boldsymbol{v} \cdot \nabla_y \theta$ is the unique minimizer over H^1_Y, which proves the Lemma. ■

With the projection $\mathcal{Q}$ defined, we immediately have from Lemmas 4.1 and 4.2 the following corollary:

Corollary 4.1 *For $u \in H^1_Y$ the projection $\mathcal{P} : H^1_Y \mapsto \mathcal{N}$ is uniquely given by $\mathcal{P}(u) = u - \mathcal{Q}(u)$.*

In order to make full use of the projections, we now present several simple but useful Lemmas.

Lemma 4.3 *$\mathcal{P}$ and $\mathcal{Q}$ are linear.*

Proof. This is obvious.

Lemma 4.4 *For $w \in \mathcal{W}$ we have $\mathcal{Q}(w) = w$.*

Proof. Since $w \in \mathcal{W}$ then $w = \boldsymbol{v} \cdot \nabla_y u$ for some $u \in H^1_Y$. Then, from Lemma 4.2 we have $\mathcal{Q}(w) = \boldsymbol{v} \cdot \nabla_y \theta$ where θ is the periodic solution of

$$\begin{aligned} \nabla_y \cdot (A \nabla_y \theta) &= \boldsymbol{v} \cdot \nabla_y w & (4.61) \\ &= \boldsymbol{v} \cdot \nabla_y (\boldsymbol{v} \cdot \nabla_y u) & (4.62) \\ &= \nabla_y \cdot (A \nabla_y u) \ . & (4.63) \end{aligned}$$

Thus, using the analysis in Lemma 4.2, $\theta = u$ uniquely up to function in $\mathcal{N}$. Then,

$$\mathcal{Q}(w) = \boldsymbol{v} \cdot \nabla_y \theta = \boldsymbol{v} \cdot \nabla_y u = w \ . \tag{4.64}$$

Lemma 4.5 *For $u \in \mathcal{N}$ we have $\mathcal{Q}(u) = 0$.*

Proof. Consider $v = u + w$ where $w \in \mathcal{W}$. Then, taking the $\mathcal{Q}$ projection gives

$$\mathcal{Q}(v) = \mathcal{Q}(u) + \mathcal{Q}(w) = \mathcal{Q}(u) + w \tag{4.65}$$

using Lemma 4.4. Then, subtracting this from $v = u + w$ and rearranging gives

$$v - \mathcal{Q}(v) - u = \mathcal{Q}(u) \ . \tag{4.66}$$

Since $v - \mathcal{Q}(v) \in \mathcal{N}$ and $u \in \mathcal{N}$ the left hand side is in $\mathcal{N}$. But $\mathcal{Q}(u) \in \mathcal{W}$ and since $\mathcal{N} \cap \mathcal{W} = \{0\}$ we must therefore have $\mathcal{Q}(u) = 0$.

Corollary 4.2 *For $u \in \mathcal{N}$ we have $\mathcal{P}(u) = u$.*

Corollary 4.3 *For $w \in \mathcal{W}$ we have $\mathcal{P}(w) = 0$.*

Lemma 4.6 *For each $u \in H^1_Y$, we have*

$$\overline{\mathcal{P}(u)} = \overline{u} \ . \tag{4.67}$$

Proof. Using the expression for $\mathcal{P}(u) = u - \mathcal{Q}(u)$ we have

$$\begin{aligned} \overline{\mathcal{P}(u)} &= \overline{u - \mathcal{Q}(u)} && (4.68) \\ &= \overline{u} - \overline{\mathcal{Q}(u)} \ . && (4.69) \end{aligned}$$

Using the expression for the projection $\mathcal{Q}(u)$ and the definition for the average gives

$$\begin{aligned} \overline{\mathcal{Q}(u)} &= \int_Y \mathcal{Q}(u) d\boldsymbol{y} && (4.70) \\ &= \int_Y \boldsymbol{v} \cdot \nabla_y \theta d\boldsymbol{y} && (4.71) \\ &= 0 && (4.72) \end{aligned}$$

since $\nabla_y \cdot \boldsymbol{v} = 0$ and thus (4.67).

Lemma 4.7 *If $u, w \in \mathcal{N}$ then $uw \in \mathcal{N}$.*

Proof. This is simply proved by expanding

$$\boldsymbol{v} \cdot \nabla_y (uw) = w(\boldsymbol{v} \cdot \nabla_y u) + u(\boldsymbol{v} \cdot \nabla_y w) = 0 \ . \tag{4.73}$$

Lemma 4.8 *For each $u \in H^1_Y$, $v \in \mathcal{N}$ we have*

$$(\mathcal{P}(u), v) = (u, v) \ . \tag{4.74}$$

Proof. From Lemma 4.1 and Lemma 4.1 we have $u = \mathcal{P}(u) + w$ with $w \in \overline{\mathcal{W}}$. Then

$$(u, v) = (\mathcal{P}(u) + w, v) = (\mathcal{P}(u), v) + (w, v) = (\mathcal{P}(u), v) \tag{4.75}$$

since $\mathcal{W} \perp \mathcal{N}$.

Lemma 4.9 *If $w \in \mathcal{N}$, then $\mathcal{P}(wv) = w\mathcal{P}(v)$ for each $v \in H^1_Y$.*

Proof. For any $u \in H^1_Y$ we have

$$(\mathcal{P}(wv), u) = (\mathcal{P}(wv), \mathcal{P}(u)) = (wv, \mathcal{P}(u)) = (v, w\mathcal{P}(u)) \ . \tag{4.76}$$

Since $w\mathcal{P}(u)$ is also in $\mathcal{N}$ by Lemma 4.7, we have

$$(v, w\mathcal{P}(u)) = (\mathcal{P}(v), w\mathcal{P}(u)) = (w\mathcal{P}(v), \mathcal{P}(u)) = (w\mathcal{P}(v), u), \tag{4.77}$$

where we have used again Lemma 4.7 to show that $w\mathcal{P}(v)$ is in $\mathcal{N}$. Thus, $(\mathcal{P}(wv), u) = (w\mathcal{P}(v), u)$ for any $u \in H^1_Y$ and the lemma follows.

Lemma 4.10 *$\mathcal{P}(u)$ and $\mathcal{Q}(u)$ are unchanged if multiply the velocity field $\boldsymbol{v}$ by $\psi \in \mathcal{N}$, $\psi \neq 0$.*

Proof. We have $\mathcal{Q}(u) = \boldsymbol{v} \cdot \nabla_y \theta$ where θ is the periodic solution of (4.48). Then, consider the projection with velocity field $\mathbf{u} = \psi \boldsymbol{v}$, i.e. $\mathcal{Q}^*(u) = \psi \boldsymbol{v} \cdot \nabla_y \theta^*$ where θ^* satisfies

$$\psi \boldsymbol{v} \cdot \nabla_y \left(\psi \boldsymbol{v} \cdot \nabla_y \theta^* \right) = \psi \boldsymbol{v} \cdot \nabla_y u \ . \tag{4.78}$$

Since $\psi \in \mathcal{N}$, $\psi \neq 0$ this gives

$$\boldsymbol{v} \cdot \nabla_y \left(\boldsymbol{v} \cdot \nabla_y (\psi \theta^*) \right) = \boldsymbol{v} \cdot \nabla_y u \ . \tag{4.79}$$

Therefore, $\psi\theta^* = \theta$ up to a function in $\mathcal{N}$. But then $\psi \boldsymbol{v} \cdot \nabla_y \theta^* = \boldsymbol{v} \cdot \nabla_y (\psi \theta^*) = \boldsymbol{v} \cdot \nabla_y \theta$ so that $\mathcal{Q}^*(u) = \mathcal{Q}(u)$. Since $\mathcal{P}(u) = u - \mathcal{Q}(u)$ this is also unchanged.

Lemma 4.11 *For $v, w \in H^1_Y$ we have*

$$(\mathcal{Q}(v), \boldsymbol{v} \cdot \nabla_y w) = (v, \boldsymbol{v} \cdot \nabla_y w) \ . \tag{4.80}$$

Proof. By simple substitution we have

$$\begin{aligned}(\mathcal{Q}(v), \boldsymbol{v}\cdot\nabla_y w) &= (v-\mathcal{P}(v), \boldsymbol{v}\cdot\nabla_y w)\\ &= (v, \boldsymbol{v}\cdot\nabla_y w) - (\mathcal{P}(v), \boldsymbol{v}\cdot\nabla_y w)\\ &= (v, \boldsymbol{v}\cdot\nabla_y w) \qquad (4.81)\end{aligned}$$

since $\mathcal{P}(v)\in\mathcal{N}$, $\boldsymbol{v}\cdot\nabla_y w\in\mathcal{W}$ and $\mathcal{W}\perp\mathcal{N}$.

Lemma 4.12 *If $u\in\mathcal{N}$ then*

$$(\mathcal{Q}(\partial_{x_i}u), \boldsymbol{v}\cdot\nabla_y w) = -(u, \partial_{x_i}(\boldsymbol{v})\cdot\nabla_y w) \qquad (4.82)$$

and

$$(\mathcal{Q}(\partial_t u), \boldsymbol{v}\cdot\nabla_y w) = -(u, \partial_t(\boldsymbol{v})\cdot\nabla_y w) \qquad (4.83)$$

holds for all $w\in H^1_Y$.

Proof. In Lemma 4.11 let $v=\frac{\partial u}{\partial x_1}$, where $u\in\mathcal{N}$. By definition, we have $\boldsymbol{v}\cdot\nabla_y u=0$, and

$$\frac{\partial}{\partial x_1}(\boldsymbol{v}\cdot\nabla_y u) = \frac{\partial v_i}{\partial x_1}\frac{\partial u}{\partial y_i} + v_i\frac{\partial^2 u}{\partial x_1\partial y_i} = 0\ . \qquad (4.84)$$

Thus, we have

$$\frac{\partial v_i}{\partial x_1}\frac{\partial u}{\partial y_i} = -v_i\frac{\partial^2 u}{\partial x_1\partial y_i}\ . \qquad (4.85)$$

Now using Lemma 4.11 with $\mathcal{Q}(\partial_{x_1}u)$ we get

$$\begin{aligned}\left(\mathcal{Q}\left(\frac{\partial u}{\partial x_1}\right), \boldsymbol{v}\cdot\nabla_y w\right) &= \left(\frac{\partial u}{\partial x_1}, \boldsymbol{v}\cdot\nabla_y w\right) &(4.86)\\ &= \left(\frac{\partial u}{\partial x_1}, \nabla_y\cdot(\boldsymbol{v}w)\right) &(4.87)\\ &= \int_Y \frac{\partial u}{\partial x_1}\frac{\partial}{\partial y_i}(v_i w)d\boldsymbol{y} &(4.88)\\ &= -\int_Y v_i w\frac{\partial^2 u}{\partial x_1\partial y_i}d\boldsymbol{y} &(4.89)\\ &= \int_Y w\frac{\partial v_i}{\partial x_1}\frac{\partial u}{\partial y_i}d\boldsymbol{y} &(4.90)\end{aligned}$$

using (4.85). Integration by parts on this gives us

$$-\int_Y u\frac{\partial}{\partial y_i}\left(w\frac{\partial v_i}{\partial x_1}\right)d\boldsymbol{y} = -\int_Y u\left[\frac{\partial v_i}{\partial x_1}\frac{\partial w}{\partial y_i}+w\frac{\partial}{\partial x_1}\left(\frac{\partial v_i}{\partial y_i}\right)\right]d\boldsymbol{y} \tag{4.91}$$

$$= -\int_Y u\left(\frac{\partial v_i}{\partial x_1}\frac{\partial w}{\partial y_i}\right)d\boldsymbol{y} \tag{4.92}$$

using the fact that $\nabla_y \cdot \boldsymbol{v} = 0$. Thus we obtain the Lemma. The other results are derived in an exactly similar manner. ■

Lemma 4.12 is very useful since it provides an alternative means of calculating the quantity $\mathcal{Q}(\partial_t u)$, which can be seen from the following lemma:

Lemma 4.13 *For $u \in H^1_Y$ the projection $\mathcal{Q}(\partial_t u)$ can be uniquely determined by $\mathcal{Q}(\partial_t u) = \boldsymbol{v}\cdot\nabla_y\phi$ where ϕ is the solution of the degenerate elliptic PDE*

$$\nabla_y \cdot (A\nabla_y\phi) = -\frac{\partial \boldsymbol{v}}{\partial t}\cdot\nabla_y u \tag{4.93}$$

with periodic boundary conditions where A is the 2×2 matrix with components $A_{ij} = v_i v_j$.

Proof. From the proof of Lemma 4.2 we know that the equation (4.93) has a solution which is unique up to a function in $\mathcal{N}$. For all $w \in H^1_Y$ we have, using the definition for ϕ that

$$(\boldsymbol{v}\cdot\nabla_y\phi, \boldsymbol{v}\cdot\nabla_y w) = \int_Y v_i v_j \frac{\partial\phi}{\partial y_i}\frac{\partial w}{\partial y_j}d\boldsymbol{y} \tag{4.94}$$

$$= -\int_Y w\frac{\partial}{\partial y_i}\left(v_i v_j\frac{\partial\phi}{\partial y_j}\right)d\boldsymbol{y} \tag{4.95}$$

$$= \int_Y w\frac{\partial v_i}{\partial t}\frac{\partial u}{\partial y_i}d\boldsymbol{y} \tag{4.96}$$

$$= \int_Y w\frac{\partial}{\partial y_i}\left(u\frac{\partial v_i}{\partial t}\right)d\boldsymbol{y} \tag{4.97}$$

$$= -\int_Y u\frac{\partial v_i}{\partial t}\frac{\partial w}{\partial y_j}d\boldsymbol{y} \tag{4.98}$$

$$= -(u, \partial_t(\boldsymbol{v})\cdot\nabla_y w)\ . \tag{4.99}$$

From Lemma 4.12 we also have that

$$(\mathcal{Q}(\partial_t u), \boldsymbol{v}\cdot\nabla_y w) = -(u, \partial_t(\boldsymbol{v})\cdot\nabla_y w) \tag{4.100}$$

holds for all $w \in H^1_Y$. Since $\boldsymbol{v}\cdot\nabla_y w$ spans $\overline{\mathcal{W}}$, we therefore have that $\mathcal{Q}(\partial_t u) = \boldsymbol{v}\cdot\nabla_y\phi$

uniquely determines the projection. ■

An exactly similar result holds for the projection $\mathcal{Q}(\partial_{x_i} u)$. From the above lemma, we see that $\mathcal{Q}(\partial_t u)$ can be found without explicitly calculating $\partial_t u$. This will be useful in the development of a numerical scheme later on.

Another, more intuitively meaningful, form of the projection $\mathcal{P}$ can be derived. From Lemma 4.1 and equation (4.48) we have $\mathcal{P}(u) = u - \boldsymbol{v} \cdot \nabla_y \theta$ where θ is the solution of (4.48). In this equation the matrix A is symmetric and therefore we can write it in the diagonal form $A = TDT^T$ where T is an orthonormal matrix

$$T = \frac{1}{\sqrt{v_1^2 + v_2^2}} \begin{pmatrix} v_1 & v_2 \\ v_2 & -v_1 \end{pmatrix}, \quad D = \begin{pmatrix} v_1^2 + v_2^2 & 0 \\ 0 & 0 \end{pmatrix} . \tag{4.101}$$

Note that A is singular and hence D has only a single no-zero diagonal element. Now introduce a new set of coordinates $\tilde{y}$ such that $\nabla_{\tilde{y}} = T \nabla_y$. Then, equation (4.48) can be written using the $\tilde{y}$ variables as

$$\nabla_{\tilde{y}} \cdot \left(D \nabla_{\tilde{y}} \tilde{\theta} \right) = \boldsymbol{v} \cdot \left(T^{-1} \nabla_{\tilde{y}} \tilde{u} \right) \tag{4.102}$$

where $\tilde{\theta}(\tilde{y}_1, \tilde{y}_2) = \theta(y_1, y_2)$ and similarly for $\tilde{u}$. Expanding and simplifying the right-hand side and also using the form of the matrix D we obtain the much simpler equation

$$\frac{\partial}{\partial \tilde{y}_1} \left[(v_1^2 + v_2^2) \frac{\partial \tilde{\theta}}{\partial \tilde{y}_1} \right] = \sqrt{v_1^2 + v_2^2} \frac{\partial \tilde{u}}{\partial \tilde{y}_1} \tag{4.103}$$

which contains only the $\tilde{y}_1$-derivatives. Solving for $\tilde{\theta}$ from this gives

$$\tilde{\theta} = \int_0^{\tilde{y}_1} \frac{1}{v_1^2 + v_2^2} \int_0^{\eta} \sqrt{v_1^2 + v_2^2} \frac{\partial \tilde{u}}{\partial \tilde{y}_1} d\xi d\eta + c \int_0^{\tilde{y}_1} \frac{1}{v_1^2 + v_2^2} d\eta + d \tag{4.104}$$

where $c = c(\tilde{y}_2, t)$ and $d = d(\tilde{y}_2, t)$ are to be determined, and η is the dummy variable for $\tilde{y}_1$ in the integration. From Lemma 4.2, the projection is computed as $\mathcal{Q}(u) = \boldsymbol{v} \cdot \nabla_y \theta$. and in the new coordinates ($\tilde{y}_1$, $\tilde{y}_2$), using (4.104) this gives

$$\mathcal{Q}(u) = \sqrt{v_1^2 + v_2^2} \frac{\partial \tilde{\theta}}{\partial \tilde{y}_1} = \frac{1}{\sqrt{v_1^2 + v_2^2}} \int_0^{\tilde{y}_1} \sqrt{v_1^2 + v_2^2} \frac{\partial \tilde{u}}{\partial \tilde{y}_1} d\eta + \frac{c}{\sqrt{v_1^2 + v_2^2}} . \tag{4.105}$$

The constants in (4.104) are determined from the boundary conditions. However, these

are not immediately obvious in the coordinates $(\tilde{y}_1,\ \tilde{y}_2)$. To determine the corresponding boundary conditions we first derive expressions for $\tilde{y}_1,\ \tilde{y}_2$. To do this, use $\nabla_y = T\nabla_{\tilde{y}}$ and apply this to $\tilde{y}_1$ and $\tilde{y}_2$ to obtain the equations:

$$\frac{\partial \tilde{y}_1}{\partial y_1} = \frac{1}{\sqrt{v_1^2+v_2^2}}\left(v_1 + v_2\frac{\partial \tilde{y}_1}{\partial \tilde{y}_2}\right), \tag{4.106}$$

$$\frac{\partial \tilde{y}_1}{\partial y_2} = \frac{1}{\sqrt{v_1^2+v_2^2}}\left(v_2 - v_1\frac{\partial \tilde{y}_1}{\partial \tilde{y}_2}\right), \tag{4.107}$$

$$\frac{\partial \tilde{y}_2}{\partial y_1} = \frac{1}{\sqrt{v_1^2+v_2^2}}\left(v_1\frac{\partial \tilde{y}_2}{\partial \tilde{y}_1} + v_2\right), \tag{4.108}$$

$$\frac{\partial \tilde{y}_2}{\partial y_2} = \frac{1}{\sqrt{v_1^2+v_2^2}}\left(v_2\frac{\partial \tilde{y}_2}{\partial \tilde{y}_1} - v_1\right). \tag{4.109}$$

Eliminating the cross-derivative terms $\frac{\partial \tilde{y}_1}{\partial \tilde{y}_2}$ and $\frac{\partial \tilde{y}_2}{\partial \tilde{y}_1}$ gives the equations

$$\boldsymbol{v}\cdot\nabla_y\tilde{y}_1 = \sqrt{v_1^2+v_2^2}, \tag{4.110}$$

$$\boldsymbol{v}^{\perp}\cdot\nabla_y\tilde{y}_2 = \sqrt{v_1^2+v_2^2}, \tag{4.111}$$

where $\boldsymbol{v}^{\perp} = (-v_2, v_1)$. Dividing both sides of equation (4.110) by $\sqrt{v_1^2+v_2^2}$ we have

$$\frac{d\tilde{y}_1}{d\boldsymbol{n}} = 1 \tag{4.112}$$

where $\boldsymbol{n} = \frac{\boldsymbol{v}}{|\boldsymbol{v}|}$ is the unit vector tangential to the streamline. Thus, the interpretation of $\tilde{y}_1$ is that it is the arc-length along a streamline of the flow. Similarly, at each point on the streamline $\tilde{y}_2$ is increasing in the orthogonal direction (it is trivial to show that $\tilde{y}_1,\ \tilde{y}_2$ form an orthogonal coordinate system). Since in the original coordinates $y_1,\ y_2$, the flow is periodic, the streamlines will either "wrap-around" in a finite number of times or else never reconnect. Consider the first case; since the streamlines reconnect, all smooth, continuous functions on this trajectory must be periodic. Denote this period by P. Then, applying this periodic boundary condition to (4.104) gives us

$$c = -\frac{\displaystyle\int_0^P \frac{1}{v_1^2+v_2^2}\int_0^{\eta}\sqrt{v_1^2+v_2^2}\frac{\partial \tilde{u}}{\partial \tilde{y}_1}d\xi d\eta}{\displaystyle\int_0^P \frac{1}{v_1^2+v_2^2}d\eta} \tag{4.113}$$

and d will be an arbitrary constant.

Now consider a particular fluid particle on the streamlines $\boldsymbol{z}(\tau)$, i.e. a *Lagrangian* description, with coordinates $(\tilde{y}_1, \tilde{y}_2)$. In view of the fact that $\tilde{y}_1$ is the arc-length along a streamline this means that we must have

$$\frac{d\tilde{y}_1}{d\tau} = \sqrt{v_1^2 + v_2^2} \ . \tag{4.114}$$

Further, assume that $\sqrt{v_1^2 + v_2^2}$ is independent of τ, which is reasonable in light of the results of section (3.3) which showed that the veloctity field and streamlines are slowly varying with time. Then, we can use this to simplify (4.105) and (4.113). From (4.114) we get

$$\tilde{y}_1 = \tau\sqrt{v_1^2 + v_2^2} + c_1 \tag{4.115}$$

where c_1 is a constant, which we can set to zero by choice the fast-time parametrization along the particular streamline. Changing integration variables to τ in (4.105) using

$$d\tilde{y}_1 = \sqrt{v_1^2 + v_2^2}d\tau \tag{4.116}$$

$$\frac{\partial}{\partial\tilde{y}_1} = \frac{1}{\sqrt{v_1^2 + v_2^2}}\frac{\partial}{\partial\tau} \tag{4.117}$$

gives us

$$\mathcal{Q}(u) = \tilde{u}(\boldsymbol{x}, \tilde{y}_1(\tau), \tilde{y}_2, t, \tau) - \frac{1}{T}\int_0^T \tilde{u}(\boldsymbol{x}, \tilde{y}_1(\tau), \tilde{y}_2, t, \tau)d\tau \tag{4.118}$$

where $T = \sqrt{v_1^2 + v_2^2}P$. Equivalently, we have in our original coordinate system

$$\mathcal{Q}(u) = u - \frac{1}{T}\int_0^T u(\boldsymbol{x}, \boldsymbol{z}(\tau), t, \tau)d\tau \tag{4.119}$$

where

$$\frac{d\boldsymbol{z}}{d\tau} = \boldsymbol{v} \ , \tag{4.120}$$

since $\tilde{y}_1$ and $\tilde{y}_2$ are the coordinates along the streamline.

Now using Corollary 4.1 we obtain

$$\mathcal{P}(u) = \frac{1}{T}\int_0^T u(\boldsymbol{x}, \boldsymbol{z}(\tau), t, \tau)d\tau \tag{4.121}$$

with $\boldsymbol{z}(\tau)$ again given by (4.120).

Thus, from the above analysis we have the following equivalent definition of the projec-

tion $\mathcal{P}$:

Lemma 4.14 *The projection $\mathcal{P} : H^1_Y \mapsto \mathcal{N}$ is uniquely given by*

$$\mathcal{P}(u)(\boldsymbol{x}, \boldsymbol{y}, t) = \lim_{T\to\infty} \frac{1}{T} \int_0^T u(\boldsymbol{x}, \boldsymbol{\Theta}(s), t, s) ds \tag{4.122}$$

where $\boldsymbol{\Theta}(\boldsymbol{x}, t, \tau; \boldsymbol{y})$ is the flow map defined by

$$\frac{d\boldsymbol{\Theta}}{d\tau} = \boldsymbol{v}, \ \boldsymbol{\Theta}(0) = \boldsymbol{y} \ . \tag{4.123}$$

The interpretation of the projection $\mathcal{P}(u)$ is now obvious. It is the average of the quantity u along the streamlines and we therefore will refer to it as the *streamline* averaging. It is the natural complement to the spatial average for this problem: the spatial average eliminates dependence on the fast-spatial scales; the streamline average eliminates dependence on the fast-time scales.

Consider again our set of equations from the multi-scale expansion. From the $O(\epsilon^{-1})$ equation

$$\boldsymbol{v} \cdot \nabla_y f' = 0 \tag{4.124}$$

we have $f' \in \mathcal{N}$. Recalling that f' is determined from $f' = f(\overline{S} + S') - \overline{f}$ and that $f(S)$ is smooth and $\overline{f}$ is independent of $\boldsymbol{y}$ we have

$$\boldsymbol{v} \cdot \nabla_y \left[f(\overline{S} + S') - \overline{f} \right] = f_S \boldsymbol{v} \cdot \nabla_y S' \tag{4.125}$$

where

$$f_S = \left. \frac{df}{dS} \right|_{\overline{S}+S'} \ . \tag{4.126}$$

Thus, from this we see that we have $S' \in \mathcal{N}$ provided that $f_S \neq 0$. From (2.56) $f(S)$ is given by $f(S) = \frac{S^2}{S^2 + a(1-S)^2}$ with $a > 0$ so that

$$\frac{df}{dS} = \frac{2aS(1-S)}{(S^2 + a(1-S)^2)^2} \ . \tag{4.127}$$

From this we see that (4.126) is zero for $\overline{S} + S' = 0, 1$ only. However, note that if $\overline{S} = 1$ then this implies that $S' \equiv 0$ in the cell and therefore $\boldsymbol{v} \cdot \nabla_y S' = 0$. Similarly for the case of $\overline{S} = 0$. If $\overline{S} \neq 0, 1$ and $\overline{S} + S' = 1$ then we must clearly have $\frac{\partial S'}{\partial y_1} = 0$ and $\frac{\partial S'}{\partial y_2} = 0$

(since it is a maximum) and hence $\boldsymbol{v} \cdot \nabla_y S' = 0$. The same argument holds for the case where $\overline{S} + S' = 0$ and where S' must be a minimum. Thus, we can conclude that $S' \in \mathcal{N}$ everywhere.

Equation (4.125) only provides a constraint that $S' \in \mathcal{N}$ but S' cannot be solved for directly from this equation. In order to determine S' we will need to develop a second equation that describes its evolution in time.

Now consider the $O(\epsilon^0)$ equation. Taking the spatial average of this equation and using that fact that all fluctuating terms have mean zero gives us, upon rearrangement:

$$\frac{\partial \overline{S}}{\partial t} + \overline{\boldsymbol{v}} \cdot \nabla_x \overline{f} = -\nabla_x \cdot \overline{\boldsymbol{v}' f'} \ . \tag{4.128}$$

This equation is basically similar to our original equation (4.32) since we have $\nabla_x \cdot \overline{\boldsymbol{v}} = 0$ and so the homogeneous part gives a conservation law for $\overline{S}$. The right-hand side term corresponds to the interaction of the small scale fluctuations upon the large scale average. The overall nature of this equation is not immediately clear without knowledge of the flux fluctuation f'. The essence of the upscaling problem is how to accurately compute this term without computing the actual fluctuations S' at all points. Towards this end, we first derive the equation for S'. Subtract (4.128) from equation (4.41) to obtain:

$$\frac{\partial S'}{\partial t} + \boldsymbol{v}' \cdot \nabla_x \overline{f} + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_x f' + \frac{\partial S_1}{\partial \tau} + (\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_y f_1 - \nabla_x \cdot \overline{\boldsymbol{v}' f'} = 0 \ . \tag{4.129}$$

We now apply the $\mathcal{P}$ projection to this equation. Consider each of the terms: for the first term we have

$$\mathcal{P}\left(\frac{\partial S'}{\partial t}\right) = \frac{\partial S'}{\partial t} - \mathcal{Q}\left(\frac{\partial S'}{\partial t}\right) \ . \tag{4.130}$$

The second term on the right-hand side is computable without having to evaluate $\frac{\partial S'}{\partial t}$ if we use Lemma 4.13 since $S' \in \mathcal{N}$ (only knowledge of S' and $\frac{\partial S'}{\partial t}$ is needed). This is useful since we obtain the time derivative in explicit form. Next, using $\frac{\partial \overline{f}}{\partial x_i} \in \mathcal{N}$ (since it has no y-dependence) and Lemma 4.9 we have

$$\mathcal{P}\left(\boldsymbol{v}' \cdot \nabla_x \overline{f}\right) = \mathcal{P}\left(\boldsymbol{v}'\right) \cdot \nabla_x \overline{f} \tag{4.131}$$

where $\mathcal{P}(\boldsymbol{v}') = (\mathcal{P}(v_1), \mathcal{P}(v_2))$. Similarly, we have

$$\mathcal{P}\left(\nabla_x \cdot \overline{\boldsymbol{v}' f'}\right) = \nabla_x \cdot \overline{\boldsymbol{v}' f'} \ . \tag{4.132}$$

since there is no y-dependence. The projection of the other terms in the equation are more complicated to evaluate. First consider the third term of (4.129)

$$\mathcal{P}\left((\overline{\boldsymbol{v}} + \boldsymbol{v}') \cdot \nabla_x f'\right) = \mathcal{P}\left(\overline{\boldsymbol{v}} \cdot \nabla_x f'\right) + \mathcal{P}\left(\boldsymbol{v}' \cdot \nabla_x f'\right) \ . \tag{4.133}$$

For the first of the term in (4.133),

$$\begin{aligned} \mathcal{P}\left(\overline{\boldsymbol{v}} \cdot \nabla_x f'\right) &= \overline{\boldsymbol{v}} \cdot \mathcal{P}\left(\nabla_x f'\right) & (4.134)\\ &= \overline{\boldsymbol{v}} \cdot \nabla_x f' - \overline{\boldsymbol{v}} \cdot \mathcal{Q}\left(\nabla_x f'\right) \ . & (4.135) \end{aligned}$$

For the second term in (4.133) we have

$$\begin{aligned} \mathcal{P}\left(\boldsymbol{v}' \cdot \nabla_x f'\right) &= \mathcal{P}\left(\left(\mathcal{P}(\boldsymbol{v}') + \mathcal{Q}(\boldsymbol{v}')\right) \cdot \nabla_x f'\right) & (4.136)\\ &= \mathcal{P}\left(\mathcal{P}(\boldsymbol{v}') \cdot \nabla_x f'\right) + \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \nabla_x f'\right) \ . & (4.137) \end{aligned}$$

For the first term in (4.137) we use Lemma 4.9 to obtain

$$\begin{aligned} \mathcal{P}\left(\mathcal{P}(\boldsymbol{v}') \cdot \nabla_x f'\right) &= \mathcal{P}(\boldsymbol{v}') \cdot \mathcal{P}\left(\nabla_x f'\right) & (4.138)\\ &= \mathcal{P}(\boldsymbol{v}') \cdot \nabla_x f' - \mathcal{P}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right) & (4.139) \end{aligned}$$

and for the second term in (4.137) we have

$$\begin{aligned} \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \nabla_x f'\right) &= \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \left(\mathcal{P}\left(\nabla_x f'\right) + \mathcal{Q}\left(\nabla_x f'\right)\right)\right) & (4.140)\\ &= \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{P}\left(\nabla_x f'\right)\right) + \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) & (4.141)\\ &= \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) \ . & (4.142) \end{aligned}$$

Thus, we obtain

$$\mathcal{P}\left((\overline{\boldsymbol{v}}+\boldsymbol{v}')\cdot\nabla_x f'\right)=\left(\overline{\boldsymbol{v}}+\mathcal{P}(\boldsymbol{v}')\right)\cdot\nabla_x f'-\left(\overline{\boldsymbol{v}}+\mathcal{P}(\boldsymbol{v}')\right)\cdot\mathcal{Q}\left(\nabla_x f'\right)+\mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}')\cdot\mathcal{Q}\left(\nabla_x f'\right)\right)\ . \tag{4.143}$$

Now consider the projection of the remaining terms which involve the fast time τ:

$$\mathcal{P}\left(\frac{\partial S_1}{\partial\tau}+(\overline{\boldsymbol{v}}+\boldsymbol{v}')\cdot\nabla_y f_1\right)\ . \tag{4.144}$$

From (4.36) we have $f_1=f_S S_1$. Note that $\boldsymbol{v}\cdot\nabla_y f_S=0$ since $f_S=\frac{df}{dS}|_{\overline{S}+S'}$ and therefore $\boldsymbol{v}\cdot\nabla_y S'=0$ ($f_{SS}=0$ only at values of S less than the shock height) Thus,

$$\frac{\partial S_1}{\partial\tau}+(\overline{\boldsymbol{v}}+\boldsymbol{v}')\cdot\nabla_y f_1=\frac{\partial S_1}{\partial\tau}+f_S(\overline{\boldsymbol{v}}+\boldsymbol{v}')\cdot\nabla_y S_1\ . \tag{4.145}$$

If we project the right-hand side of this equation onto streamlines $\tilde{\boldsymbol{z}}$ defined the velocity field $f_S\boldsymbol{v}$, then this becomes the total derivative $\frac{dS_1}{d\tilde{\boldsymbol{z}}}$. By Lemma 4.10, the projection $\mathcal{P}$ is unchanged by multiplying the velocity field by a function $g\in\mathcal{N}$. Therefore, using the alternative form of the projection, with $f_S\boldsymbol{v}$ instead of $\boldsymbol{v}$ we get

$$\begin{aligned}\mathcal{P}\left(\frac{\partial S_1}{\partial\tau}+f_S(\overline{\boldsymbol{v}}+\boldsymbol{v}')\cdot\nabla_y S_1\right) &= \lim_{T\to\infty}\frac{1}{T}\int_0^T\frac{dS_1}{d\tilde{\boldsymbol{z}}}d\tau &(4.146)\\ &= \lim_{T\to\infty}\frac{S_1(T)-S_1(0)}{T} &(4.147)\\ &= 0 &(4.148)\end{aligned}$$

if S_1 is bounded.

Combining the above results, we obtain the following theorem:

Theorem 4.2 *For the ansatz (4.33), $\overline{S}$ and S' satisfy the following closed, coupled system of equations*

$$\begin{aligned}\frac{\partial\overline{S}}{\partial t}+\overline{\boldsymbol{v}}\cdot\nabla_x\overline{f}+\nabla_x\cdot\overline{\boldsymbol{v}'f'} &= 0\ , &(4.149)\\ \frac{\partial S'}{\partial t}+\left(\overline{\boldsymbol{v}}+\mathcal{P}(\boldsymbol{v}')\right)\cdot\nabla_x f'+\mathcal{P}\left(\boldsymbol{v}'\right)\cdot\nabla_x\overline{f}-\nabla_x\cdot\overline{\boldsymbol{v}'f'} &= G(\boldsymbol{x},\boldsymbol{y},t) &(4.150)\end{aligned}$$

where

$$G(\boldsymbol{x},\boldsymbol{y},t) = \left(\overline{\boldsymbol{v}} + \mathcal{P}(\boldsymbol{v}')\right)\cdot\mathcal{Q}\left(\nabla_x f'\right) - \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}')\cdot\mathcal{Q}\left(\nabla_x f'\right)\right) + \mathcal{Q}\left(\frac{\partial S'}{\partial t}\right) . \tag{4.151}$$

Furthermore, for each fixed $\boldsymbol{y}$ the system is hyperbolic with respect to the variables $\boldsymbol{x}$ and t.

Proof. Combining the previous results gives us the form of the equations. We therefore only need to demonstrate that the system is hyperbolic in the variables $\boldsymbol{x}$ and t (note that the fast spatial variable $\boldsymbol{y}$ now appears only as a parameter in the above system and also that the fast time τ has been completely eliminated). Hyperbolicity is proved in Appendix A. ■

Note that the our system (4.149) and (4.150) is not in conservation form, even though the original equation (4.32) defines a conservation law. This is due to the fact that the original saturation has been split as $S_\epsilon = \overline{S} + S' + O(\epsilon)$.

4.4.1 Justification for the Asymptotic Expansions

In our analysis we been deliberately vague with the choice of initial conditions for the terms in the expansion of the saturation. This is due to the fact that terms involving the fast time scale τ appear in the analytic solution do not generally appear if we start from smooth initial data (i.e. the initial saturation is a function of the large scale $\boldsymbol{x}$ only). To see this, we write the solution of (4.32) as $S_\epsilon = \hat{S}(\boldsymbol{x},\boldsymbol{y},t) + \tilde{S}(\boldsymbol{x},\boldsymbol{y},t,\tau)$, i.e. an "average" with respect to the fast time plus a fluctuation about this average. Then, substituting into (4.32), using (4.38) and (4.39) and gathering terms with the same power of ϵ gives at $O(\epsilon^{-1})$

$$\frac{\partial \tilde{S}}{\partial \tau} + \boldsymbol{v}\cdot\nabla_y\tilde{S} + \boldsymbol{v}\cdot\nabla_y\hat{S} = 0 . \tag{4.152}$$

If S_ϵ is initially smooth then $\boldsymbol{v}\cdot\nabla_y\hat{S}$ will be zero and $\tilde{S} = 0$. Hence $\tilde{S}$ will be identically zero for all subsequent times and hence no fast time scales appear in the solution. Note that this is also the case if $\hat{S} \in \mathcal{N}$ initially (this provides the constraint on the initial form of the fluctuations S'). If $\hat{S}$ has any component in $\mathcal{W}$ then $\tilde{S}$ will be non-zero and hence the fast time scale appears. In our problems, the initial saturation will always be smooth. However, in the course of numerical computations, at steps beyond the first, we are in effect solving (4.32) with oscillatory initial data and due to numerical errors this may not exactly

lie in the space $\mathcal{N}$. Therefore, it is important to show that if this is the case, these errors do not grow.

To show that this is indeed the case, we need to show that S_1 remains bounded as $\tau \to \infty$. To do this, we derive the equation for S_1. Taking (4.150) as our *given* equation for S', we subtract it from the fluctuation equation (4.129) to obtain, upon simplification,

$$\frac{\partial S_1}{\partial \tau} + f_S \boldsymbol{v} \cdot \nabla_y S_1 = -\mathcal{Q}(\boldsymbol{v}') \cdot \left[\nabla_x (\overline{f} + f')\right] - \left(\overline{\boldsymbol{v}} + \mathcal{P}(\mathbf{v}')\right) \cdot \mathcal{Q}(\nabla_x f') + \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) - \mathcal{Q}\left(\frac{\partial S'}{\partial t}\right) . \tag{4.153}$$

If we project this onto the streamlines defined by $f_S \boldsymbol{v}$ then the left-hand side becomes a total derivative. To show that S_1 remains bounded, we must estimate how fast the terms on the right-hand side decay along the streamline. Suppose first that the streamlines reconnect. Then, by Lemma 4.10, since the projection $\mathcal{P}$ is invariant if we multiply $\boldsymbol{v}$ by $g \in \mathcal{N}$, the integral of right-hand side over one such period is exactly $P\mathcal{P}(RHS)$, where as before P is the length of the path the streamlines traverse before reconnecting. Taking the $\mathcal{P}$ projection of these terms and using the fact that $\mathcal{P}(\mathcal{Q}(u)) = 0$ and the other properties of $\mathcal{P}$ these become

$$-\mathcal{P}\left(\mathcal{Q}(\mathbf{v}') \cdot \nabla_x f'\right) - \mathcal{P}\left(\mathcal{P}(\boldsymbol{v}') \cdot \mathcal{Q}(\nabla_x f')\right) + \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) . \tag{4.154}$$

Using Lemma 4.9 the second term is zero. Then, combining the other terms

$$\begin{aligned} -\mathcal{P}\left(\mathcal{Q}(\mathbf{v}') \cdot \nabla_x f'\right) + \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) &= \mathcal{P}\left[\mathcal{Q}(\boldsymbol{v}') \cdot \left(\nabla_x f' - \mathcal{Q}\left(\nabla_x f'\right)\right)\right] && (4.155)\\ &= \mathcal{P}\left[\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{P}(\nabla_x f')\right] && (4.156)\\ &= \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}')\right) \cdot \mathcal{P}(\nabla_x f') && (4.157)\\ &= 0 . && (4.158) \end{aligned}$$

Thus, $P\mathcal{P}(RHS) = 0$. Hence Hence S_1 is periodic and bounded over this interval P.

If the streamlines do not reconnect, then we still have that the average of the right-hand side terms approach zero as $T \to \infty$. Thus, we have $S_1(T)/T \to 0$ as $T \to \infty$ which shows that S_1 at least grows sub-linearly.

Chapter 5

Numerical Implementation

5.1 Overview

In this section of this thesis we describe how to take the analysis given in the first previous sections and translate this into a scheme for computing upscaled numerical solutions to our two-phase flow problem.

The results of the analysis given in Chapter 2 lead to the upscaled equations (4.24) for the pressure equation, and (4.149) for the saturation. As mentioned, both retain the original character of the problem (2.51), (2.52), i.e. the upscaled pressure equation remains elliptic and the upscaled saturation equation remains hyperbolic. From now on, when we refer to the "saturation" equation we mean the upscaled equation (4.149) and when we refer to the "pressure" equation we mean (4.24). The multiple scale analysis was sequential, in that period fluctuations in the permeability give rise to period fluctuations in the velocity and this then gives rise to periodic fluctuations in the saturation. The numerical method we employ is similarly sequential, in the same way that the resolved computations were (an IMPES scheme). We solve the pressure equation (4.24) via an implicit, ellptic method and then use the resulting velocity field to explicitly update the saturation (4.149). These equations are solved on *coarse* grids. However, as was noted, in both equations we need to compute fluctuating quantities (from (4.25) for the pressure equation and (4.150) and the velocity fluctuations for the saturation equation). Thus, in addition to the coarse grid, we also define sub-grids within each of the coarse grid cells that enable us to compute these quantities.

We first concentrate on the numerical method for solving (4.149), which is the main emphasis of this thesis. Combining this with equation (4.150) leads to a coupled system

of hyperbolic equations with source terms. These are solved using a finite-volume method. This scheme for the saturation equation then guides us in choosing a numerical method for (4.24). We are able to use a variant of the multi-scale finite element method for this. These two numerical methods complement one another very nicely to give a succicnt upscaling method.

This chapter is laid out as follows. We first describe the overall strategy for solving the saturation equation and how this leads to what we call the coarse-grid sub-grid method. We present the details of the numerical method, the hyperbolic solver and the method used to compute the streamline average (4.122). At that point we then demonstrate the converge of the scheme for this part by performing some numerical tests. We then describe the numerical method for the pressure equation using multi-scale finite element methods, and, in particular, a special variant for periodic permeabilities that we use in our scheme. Combining the upscaling methods for the saturation equation and the pressure equation gives our overall numerical scheme. Finally, a modification is suggested that allows for the method to be used in practice where the fluctuations in the permeability are not periodic.

5.2 Coarse-grid Sub-grid Approach

Recall our set of equations for the evolution of the average and fluctuation of the saturation

$$\frac{\partial \overline{S}}{\partial t} + \overline{\boldsymbol{v}} \cdot \nabla_x \overline{f} = -\nabla_x \cdot \overline{\boldsymbol{v}' f'} \,, \tag{5.1}$$

$$\frac{\partial S'}{\partial t} + \left(\overline{\boldsymbol{v}} + \mathcal{P}(\boldsymbol{v}')\right) \cdot \nabla_x f' + \mathcal{P}\left(\boldsymbol{v}'\right) \cdot \nabla_x \overline{f} = \nabla_x \cdot \overline{\boldsymbol{v}' f'} + G(\boldsymbol{x}, \boldsymbol{y}, t) \,, \tag{5.2}$$

$\overline{S}$ being the spatial average of the multiscale solution S_ϵ (the solution of our original system (2.51), (2.52)) and S' the $O(1)$ fluctuation about this average. The terms are G given by (4.151) and note that we have now moved the small-scale large-scale interaction terms $\nabla_x \cdot \overline{\boldsymbol{v}' f'}$ to the right-hand side of both equations, where they are now treated as source terms. $\overline{\boldsymbol{v}}$ and $\boldsymbol{v}'$ are the average and fluctuation of the velocity field. Note that the fluctuation equation (5.2) has dependence on the fast-spatial variable $\boldsymbol{y}$ but that this appears only as a parameter. Thus, for each particular value $\boldsymbol{y}_{kl}$, (5.2) gives the evolution of the fluctuation

S'_{kl}, i.e.

$$\frac{\partial S'_{kl}}{\partial t} + \left(\overline{\boldsymbol{v}} + \mathcal{P}(\boldsymbol{v}')_{kl}\right) \cdot \nabla_x f'_{kl} + \mathcal{P}\left(\boldsymbol{v}'\right)_{kl} \cdot \nabla_x \overline{f} = \nabla_x \cdot \overline{\boldsymbol{v}' f'} + G(\boldsymbol{x}, \boldsymbol{y}, t)_{kl} \ , \tag{5.3}$$

with the subscript kl denoting the evaluation of terms that depend on $\boldsymbol{y}$ at the point $\boldsymbol{y}_{kl}$. Assume further that we have sufficient number of $\boldsymbol{y}_{kl}$ points to cover the cell Y, so that the average (4.6) can be approximated by

$$\begin{aligned} \overline{\phi}(\boldsymbol{x}) &= \int_Y \phi(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{y} & (5.4) \\ &\approx \sum_{k,l} h_{k,l} \phi(\boldsymbol{x}, \boldsymbol{y}_{kl}) \ , & (5.5) \end{aligned}$$

i.e. a numerical quadrature for the average. This converges as the number of points $\boldsymbol{y}_{kl}$ are increased, provided $\phi(\boldsymbol{x}, \boldsymbol{y})$ is bounded and continuous. For convenience of analysis, we can take the points $\boldsymbol{y}_{kl}$ to form a Cartesian mesh over the cell Y with K grid points along y_1 and L grid points along y_2. In this case we have $h_{k,l} = h = 1/KL$ for all k, l. The above quadrature rule is the 2-d equivalent of the trapezoidal rule, which is extremely accurate for periodic functions. Then, we can use this for the average and fluctuation fluxes and the interaction term,

$$\begin{aligned} \overline{f} &= \int_Y f(\overline{S} + S') d\boldsymbol{y} \approx \sum_{k,l} h f(\overline{S} + S'_{kl}) \ , & (5.6) \\ f' &= f(\overline{S} + S') - \overline{f} \ , & (5.7) \\ \overline{\boldsymbol{v}' f'} &= \int_Y \boldsymbol{v}' f' d\boldsymbol{y} \approx \sum_{k,l} h \boldsymbol{v}'_{kl} f'_{kl} \ . & (5.8) \end{aligned}$$

Note that from the approximation for the interaction term we can see that

$$\begin{aligned} \nabla_x \cdot \overline{\boldsymbol{v}' f'} &= \nabla_x \cdot \sum_{k,l} h \boldsymbol{v}'_{kl} f'_{kl} & (5.9) \\ &= \sum_{k,l} h \left[f'_{kl} \nabla_x \cdot \boldsymbol{v}'_{kl} + \boldsymbol{v}'_{kl} \cdot \nabla_x f'_{kl} \right] & (5.10) \\ &= \sum_{k,l} h \boldsymbol{v}'_{kl} \cdot \nabla_x f'_{kl} \ , & (5.11) \end{aligned}$$

since $\nabla_x \cdot \boldsymbol{v}' = 0$. Combining the $K \times L$ fluctuation equations (5.3) with the average equation (5.1) and using the approximation (5.11) for the term $\nabla_x \cdot \overline{\boldsymbol{v}' f'}$ leads us to the

coupled system of nonlinear equations

$$\frac{\partial \mathbf{q}}{\partial t} + A_h \frac{\partial \mathbf{f}(\mathbf{q})}{\partial x_1} + B_h \frac{\partial \mathbf{f}(\mathbf{q})}{\partial x_2} = \mathbf{H} \ , \tag{5.12}$$

where $\mathbf{q}$ is the vector

$$\mathbf{q} = \left(\overline{S}, S'_{11}, S'_{12}, \ldots, S'_{1L}, S'_{21}, S'_{22}, \ldots, S'_{KL}\right)^T \ , \tag{5.13}$$

A_h and B_h are matrices with

$$A_h = \begin{pmatrix} \overline{v}_1 & 0 & 0 & \ldots & \ldots \\ \mathcal{P}(v'_1)_{11} & \overline{v}_1 + \mathcal{P}(v'_1)_{11} & 0 & \ldots & \ldots \\ \mathcal{P}(v'_1)_{12} & 0 & \overline{v}_1 + \mathcal{P}(v'_1)_{12} & \ldots & \ldots \\ \vdots & & \ddots & & \end{pmatrix} \tag{5.14}$$

(contains only the x_1-components of the velocity field), and similarly for B_h. The h subscript is used to denote the dependence on the particular choice of the discretization of the sub-grid $\boldsymbol{y}_{kl}$. For the most part we will drop the use of this subscript for clarity. $\mathbf{f}(\mathbf{q})$ is the flux vector with components

$$\mathbf{f}(\mathbf{q}) = \left(\overline{f}, f'_{11}, f'_{12}, \ldots, f'_{1L}, f'_{21}, f'_{22}, \ldots, f'_{KL}\right)^T \tag{5.15}$$

and $\mathbf{H}$ is the vector of "source-terms"

$$\mathbf{H} = \left(-\nabla_x \cdot \overline{\boldsymbol{v}' f'}, \nabla_x \cdot \overline{\boldsymbol{v}' f'} + G_{11}, \nabla_x \cdot \overline{\boldsymbol{v}' f'} + G_{12}, \ldots, \nabla_x \cdot \overline{\boldsymbol{v}' f'} + G_{KL}\right)^T \ . \tag{5.16}$$

Equation (5.12) can be solved numerically by discretizing in $\boldsymbol{x}$, i.e. at the coarse level since the $\boldsymbol{y}$ dependence has been taken care of by our choice of $\boldsymbol{y}_{kl}$. We refer to the above methodology as coarse-grid sub-grid method. Within the domain of interest we first define a coarse-grid with a sufficient number of points $\boldsymbol{x}_{ij}$ to discretize the features on the large scale (those that vary with respect to the large scale variable $\boldsymbol{x}$). Then, within each of the coarse-grid cells we define sub-grids, the points $\boldsymbol{y}_{kl}$, to discretize the features on the fast scale $\boldsymbol{y}$. The sub-grids need not cover the whole of the coarse-grid block, nor in fact, use any particular feature of the its geometry. The sub-grid need only contain a sufficient

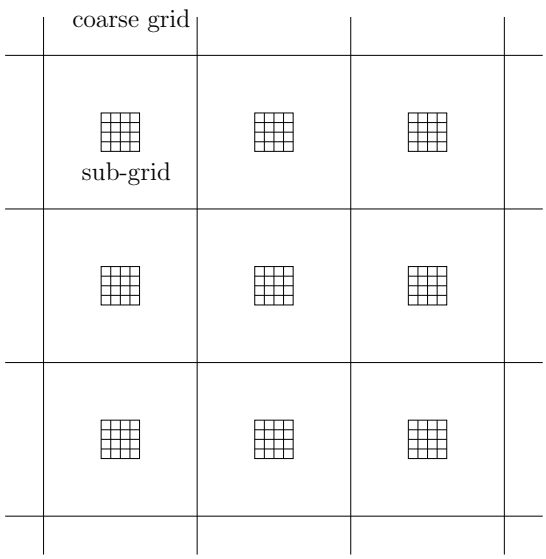


Figure 5.1: Diagram for the coarse-grid sub-grid setup.

number of points so that the integrals in (5.6) and (5.8) are computed accurately. Thus, we may choose to use a sub-grid located at the center of the coarse-grid block, as illustrated in figure 5.1. Taking this approach, we can view our sub-grid as a *sampling* of the fast-scales within the coarse-grid. Therefore, our method is similar in several respects to the work of Kevrekidis *et al* [18] who used an approach known as "patch dynamics" to develop coarse-grid models for multiscale problems. In that method a fine grid simulator is used to find the approximate evolution of the average over a short time period and then a larger step is then taken by extrapolating this forward in time. Their work, however, concentrated more on dissipative systems and also the fine scale information is not carried forward between time steps, so that the initial conditions for the fine scale solver are artificially imposed. In our case, since we retain the solution on the sub-grid between steps, we naturally use these as the initial condition for the next coarse-grid time step. Moreover, for the case of periodic velocity fields, our method exactly (to within $O(\epsilon)$) captures the true solution at each point within the domain, even if we use a sub-grid that does not cover the whole of the coarse-grid. This is because we can interpolate the solution $\overline{S} + S'$, to all points in the coarse-grid block using periodicity (strictly we must also have the coarse-grid block being an integer multiple of the fast scale period Y/ϵ).

Equation (5.12) is the basis for our numerical upscaling of the saturation equation. By solving for only a subset of the fluctuations, we hope to be able to approximate the

small scale large scale interaction term $\nabla_x \cdot \overline{\boldsymbol{v}' f'}$ sufficiently accurately so that $\overline{S}$ is evolved correctly. Since the discretization of (5.12) is done at the coarse-grid level, we benefit both from an overall smaller number of grid points than resolved calculations and the fact that we are now able to take large time steps that will still satisfy stability conditions for the numerical scheme. For the velocity field, we are able to compute the corresponding average and fluctuations on exactly the same coarse and sub-grids when we use MSFEM, as will be described in section 5.4.

We refer our system (5.12) an "upscaled" system even though we compute fluctuations which are, by definition, small scale features. Whilst this is perhaps not as pleasing as deriving a single homogenized equation for the average saturation $\overline{S}$, as can be done in the case of elliptic equations, or in special cases, it is more practical from a computational viewpoint.

5.3 Numerical Upscaling Method for the Saturation Equation

In Appendix A we show we that the homogeneous part of the upscaled equations (5.12) form a hyperbolic system of equations. The strategy we use for solving such a system when coupled with the source terms what is often called a *fractional-step* or *operator-splitting* method [24]. For this method, we split the problem (5.12) into two sub-problems; the homogeneous part

$$\frac{\partial \mathbf{q}}{\partial t} + A\frac{\partial \mathbf{f}(\mathbf{q})}{\partial x_1} + B\frac{\partial \mathbf{f}(\mathbf{q})}{\partial x_2} = 0 \tag{5.17}$$

and the source terms

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{H} \ . \tag{5.18}$$

To advance the solution forward in time we alternate between solving (5.17) and (5.18). The great advantage of this method is that we can apply existing techniques for hyperbolic equations in solving (5.17). For (5.18) we can treat these as a set of ODEs in time and so again have a wide range of solution methods at our disposal.

For the hyperbolic part, we can take advantage of the well developed theory that exists for solving such systems numerically. Further, as was shown above, the fluctuations in the saturation will develop steep gradients and shocks, consistent with the original scalar

problem. Therefore, the numerical scheme must be able to handle these features of the solution. Finite-volume schemes are perfectly suited to handle these issues and we choose to use the class of schemes known as wave-propagation methods, developed by LeVeque [24]. These are implemented via the freely available package CLAWPACK [23]. A description of the methods and their implementation for our problem is given in sections 5.3.1 and 5.3.2.

The updating of the source terms via (5.18) is done using a second order Runge-Kutta method. The reasons for this will be given in section 5.3.3.

A crucial part of the numerical implementation for the transport problem is the computation of the streamline projection and this is discussed in section 5.3.4.

Finally, in section 5.3.5 we demonstrate the convergence of the numerical scheme for the saturation part of the upscaling. The problem for the velocity field is eliminated by prescribing an analytic form for the velocity field that is divergence-free and has rapid fluctuations. By doing this we are able to show that the scheme for the hyperbolic transport part is first order overall.

5.3.1 Finite-Volume Solution of Homogeneous System

In developing a numerical scheme for (5.17), recall that the fast spatial variable $\boldsymbol{y}$ has been taken care of via our choice of the set of points $\boldsymbol{y}_{kl}$. The hyperbolic solver therefore sees the system as a function of $\boldsymbol{x}$, the coarse variable, and t only. When we refer to grids in what now follows, we mean discretizations in $\boldsymbol{x}$.

Finite volume methods are based on subdividing the spatial domain into cells (the "finite volumes") and keeping track of an approximation to the integral of $\mathbf{q}$, our vector of unknowns in the hyperbolic problem (5.17), over each of these volumes. The methods are well described in several texts, e.g. LeVeque [24]. By working with cell averages, it is easier to use important properties of the conservation law in deriving numerical methods. In particular we can make sure that the methods are conservative in a way that mimics the true solution and this is very important in accurately calculating shock waves and other features.

We will begin by considering uniform Cartesian grids, using the notation illustrated in figure 5.2. The grid point at $(x_{1,i}, x_{2,j})$ will often be abbreviated by (i, j) in this case to avoid the cumbersome use of subscripts. Each grid cell is of the form $\mathcal{C}_{i,j} = [x_{1,i-1/2}, x_{1,j+1/2}] \times [x_{2,i-1/2}, x_{2,j+1/2}]$, $\Delta x_1 = x_{1,i+1/2} - x_{1,i-1/2}$ and $\Delta x_2 = x_{2,j+1/2} - x_{2,j-1/2}$. To derive the

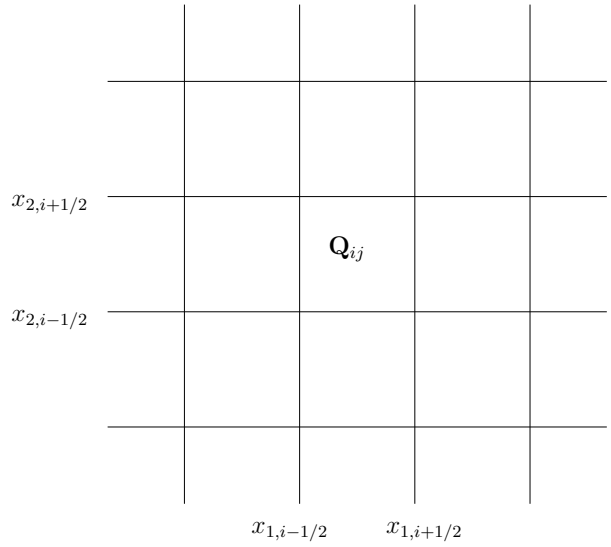


Figure 5.2: Finite volume grid in two space dimensions, where $\mathbf{Q}_{ij}$ represents cell average.

basic form of the finite-volume scheme integrate the homogeneous equation (5.17) over $\mathcal{C}_{i,j}$, which gives

$$\begin{aligned}
\frac{d}{dt}\int\int_{\mathcal{C}_{i,j}} \mathbf{q}(x_1,x_2,t)dx_1dx_2 &= -\int_{\mathcal{C}_{i,j}} A\frac{\partial \mathbf{f}}{\partial x_1}dx_1dx_2 - \int_{\mathcal{C}_{i,j}} B\frac{\partial \mathbf{f}}{\partial x_2}dx_1dx_2 \\
= \; -\int_{x_{2,j-1/2}}^{x_{2,j+1/2}} A\mathbf{f}\left(\mathbf{q}(x_{1,i+1/2},x_2,t)\right)dx_2 &+ \int_{x_{2,j-1/2}}^{x_{2,j+1/2}} A\mathbf{f}\left(\mathbf{q}x_{1,i-1/2},x_2,t)\right)dx_2 \\
-\int_{x_{1,i-1/2}}^{x_{1,i+1/2}} B\mathbf{f}\left(\mathbf{q}(x_1,x_{2,j+1/2},t)\right)dx_1 &+ \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} B\mathbf{f}\left(\mathbf{q}(x_1,x_{2,j-1/2},t)\right)dx_1 \; .
\end{aligned} \tag{5.19}$$

Now integrating this expression from t^n to $t^{n+1} = t^n + \Delta t$ and dividing by the cell area $\Delta x_1 \Delta x_2$ we obtain the fully-discrete flux-differencing method of the form

$$\mathbf{Q}_{i,j}^{n+1} = \mathbf{Q}_{i,j}^{n} - \frac{\Delta t}{\Delta x_1}\left[\mathbf{F}_{i+1/2,j}^{n} - \mathbf{F}_{i-1/2,j}^{n}\right] - \frac{\Delta t}{\Delta x_2}\left[\mathbf{G}_{i,j+1/2}^{n} - \mathbf{G}_{i,j-1/2}^{n}\right] \tag{5.20}$$

where

$$\mathbf{Q}_{i,j}^{n} \approx \frac{1}{\Delta x_1 \Delta x_2}\int_{x_{2,i-1/2}}^{x_{2,i+1/2}}\int_{x_{1,i-1/2}}^{x_{1,i+1/2}} \mathbf{q}(x_1,x_2,t)dx_1dx_2 \tag{5.21}$$

is the approximation to the average of $\mathbf{q}$ over $\mathcal{C}_{i,j}$ at time $t = t^n$ (note that it will be an

approximation since $\mathbf{q}(x_1, x_2, t)$ will not be available in general at $t = t^n$) and

$$\mathbf{F}^n_{i-1/2,j} \approx \frac{1}{\Delta t \Delta x_2} \int_{t_n}^{t_{n+1}} \int_{x_{2,i-1/2}}^{x_{2,i+1/2}} A\mathbf{f}\left(\mathbf{q}(x_{1,i-1/2}, x_2, t)\right) dx_2 dt \tag{5.22}$$

$$\mathbf{G}^n_{i,j-1/2} \approx \frac{1}{\Delta t \Delta x_1} \int_{t_n}^{t_{n+1}} \int_{x_{1,i-1/2}}^{x_{1,i+1/2}} B\mathbf{f}\left(\mathbf{q}(x_1, x_{2,i-1/2}, t)\right) dx_1 dt \tag{5.23}$$

are the numerical fluxes each each edge of the cell $\mathcal{C}_{i,j}$. Note that the averaging used here to obtain $\mathbf{Q}_{i,j}$is different that the average (4.6) used in deriving the upscaled equations. Here it is assumed that $\mathbf{q}$ has no fast spatial dependence and the average is only used for the convenience of developing a discretely conservative numerical method. Note also that the matrices A and B will be spatially varying since they depend on the velocity field.

5.3.2 Wave-Propagation Method

To obtain expressions for the fluxes (5.22), (5.23) we use the wave-propagation algorithms of LeVeque and Bale [24, 1]. These are based upon solving a Riemann problem at each cell interface between grid cells and using the resulting wave structure to update the solution in the grid cell on each side. This is also the basis for Godunov's method and other shock capturing schemes. Due to the discrete nature of the methods, it is easier to interpret the fluxes (5.22), (5.23) as updates to the cell averages, as in (5.20), rather than focusing on them as approximations to the integrals (5.22). We give a summary of the wave-propagation methods below for the 1-dimensional case. This can then be used directly for the 2-dimensional method, as in a dimension splitting method, or with some minor modifications to give an unsplit method.

First note that the system (5.17) is not autonomous, i.e. the flux function depends not just on the solution $\mathbf{q}$ but also the spatial variable $\boldsymbol{x}$. This case is not usually treated in the analysis of Godunov's method since it gives rise to a Riemann problem that is not of the classical form, but rather a *generalized* Riemann problem. The 1-dimensional problem we use to illustrate the method is

$$\mathbf{q}_t + A(x,t)\left(\mathbf{f}(\mathbf{q})\right)_{x_1} = 0 \tag{5.24}$$

which, together with

$$\mathbf{q}_t + B(x,t)\left(\mathbf{f}(\mathbf{q})\right)_{x_2} = 0 \tag{5.25}$$

is the basis for solving (5.17) by dimension splitting [24]. The matrices A and B are of the form given by (5.14). $\mathbf{f}$ is given by (5.15). For convenience, we denote by x the variable x_1 for the 1-d discussion.

The flux function $A\mathbf{f}$ is discretized with respect to x in a manner consistent with the finite-volume interpretation. Note that for our grid, two possible discretizations are possible; cell-centered flux functions or edge-centered flux functions. In a cell-centered approach the flux function $A\mathbf{f}$ is discretized to yield a flux function $(A\mathbf{f})_i(\mathbf{q})$ that holds throughout the ith grid cell (different *functional* form in each cell). This is actually the most natural approach for our problem since $\mathbf{f}$ is determined from (A.6) and (A.7) by averaging over the fast scales within a grid block. When cell-centered flux functions are used, the generalized Riemann problem at cell interface $x_{i-1/2}$ consists of the equation

$$\mathbf{q}_t + \mathbf{F}_{i-1/2}(\mathbf{q}, x)_x = 0 \tag{5.26}$$

where

$$\mathbf{F}_{i-1/2}(\mathbf{q}, x) = \begin{cases} (A\mathbf{f})_{i-1}(\mathbf{q}) & \text{if } x < x_{i-1/2} \\ (A\mathbf{f})_i(\mathbf{q}) & \text{if } x > x_{i-1/2} \end{cases} \tag{5.27}$$

together with the initial data

$$\mathbf{q}(x, 0) = \begin{cases} \mathbf{Q}_{i-1} & \text{if } x < x_{i-1/2} \\ \mathbf{Q}_i & \text{if } x > x_{i-1/2} \end{cases} \tag{5.28}$$

This Riemann problem is slightly more complicated than in the autonomous case where the equation (5.26) would be the same on either side of the discontinuity in the initial data. We give some details below. Note that this choice of discretization dictates that A, and hence the velocity field, should also be discretized in a cell-centered manner.

The alternative discretization approach is to use cell-edge flux functions. In this we assume that a distinct flux function $(A\mathbf{f})_{i-1/2}$ is associated with each cell interface $x_{i-1/2}$. This is natural if we interpret the flux function as measuring the the flow between cell $i-1$ and cell i and often makes more sense than the cell-centered approach. We can relate this to the cell-centered flux approach by viewing the flux $(A\mathbf{f})_{i-1/2}(\mathbf{q})$ as holding over the interval $[x_{i-1}, x_i]$ between the center of cell $i-1$ and the center of cell i. The Riemann problem at $x_{i-1/2}$ is now a classical Riemann problem for the single equation $\mathbf{q}_t + (A\mathbf{f})_{i-1/2}(\mathbf{q})_x = 0$

with the data (5.28). However, to be consistent it would be necessary to consider a second set of Riemann problems at the cell centers x_i, where the flux function jumps. Nontrivial waves can arise from these points even though the data $\mathbf{Q}_i$ is the same to both sides.

The basis of the numerical method is to approximately solve the generalized Riemann problem (5.26) at $x_{i-1/2}$. To motivate the method, first consider the *classical* Riemann problem for a constant-coefficient system

$$\mathbf{q}_t + C\mathbf{f}(\mathbf{q})_x = 0 \tag{5.29}$$

where the matrix C is constant, with piecewise-constant data (5.28). The solution can be expressed in terms of the eigenvectors $\mathbf{r}^p_{i-1/2}$ of Jacobian matrix $J_{i-1/2} = C\mathbf{f}'$ (note that the standard primed notation now refers to differentiation with respect to $\mathbf{q}$ rather than signifying fluctuating quantities as in the section on multiscale analysis). The standard approach is to decompose the jump in $\mathbf{Q}$ as a linear combination of the eigenvectors in order to define "waves" $\mathbf{W}^p_{i-1/2}$:

$$\mathbf{Q}_i - \mathbf{Q}_{i-1} = \sum_{p=1}^{m} \alpha^p_{i-1/2}\mathbf{r}^p_{i-1/2} \equiv \sum_{p=1}^{m} \mathbf{W}^p_{i-1/2} \ . \tag{5.30}$$

The coefficients $\alpha^p_{i-1/2}$ are given by

$$\alpha^p_{i-1/2} = R^{-1}_{i-1/2}(\mathbf{Q}_i - \mathbf{Q}_{i-1}) \tag{5.31}$$

where $R_{i-1/2}$ is the matrix of right eigenvectors. Waves corresponding to positive eigenvalues will move right into cell i and waves with positive eigenvalues will move left into cell $i-1$. Denoting the eigenvalues of $J_{i-1/2}$ by $s^p_{i-1/2}$, we would then we define fluctuations (using notation of LeVeque [24])

$$A^+\Delta\mathbf{Q}_{i-1/2} = \sum_{p=1}^{m} (s^p_{i-1/2})^+ \mathbf{W}^p_{i-1/2} \ , \tag{5.32}$$

$$A^-\Delta\mathbf{Q}_{i+1/2} = \sum_{p=1}^{m} (s^p_{i+1/2})^- \mathbf{W}^p_{i+1/2} \ . \tag{5.33}$$

Here $s^+ = \max(s, 0)$ and $s^- = \min(s, 0)$. These give, respectively, the contributions to the cell average $\mathbf{Q}_i$ due to the right-going waves from $x_{i-1/2}$ and left-going waves from $x_{i+1/2}$.

The 1-d scheme in this case is then

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x}\left[A^+\Delta\mathbf{Q}_{i-1/2} + A^-\Delta\mathbf{Q}_{i+1/2}\right] \ . \tag{5.34}$$

In this method, we have some freedom in the approximation used for the Jacobian $J_{i-1/2}$. A common choice is one such that the method is discretely conservative, i.e.

$$J_{i-1/2}(Q_i - Q_{i-1}) = C(f(Q_i - f(Q_{i-1}) \ . \tag{5.35}$$

This leads to the well-known "Roe average".

For the spatially-varying case, we need the solution of the generalized Riemann problem, (5.26), (5.27), (5.28), which is not as simple, and cannot be written in terms of simple waves as in (5.30). To see this we first consider the structure of the solution in the scalar case. This case also covers the solution of the original resolved calculations that were shown in Chapter 1. The case for systems then follows in a similar manner. Consider our original saturation equation (now using $q = S$ for the saturation),

$$q_t + (u(x,t)f(q))_x = 0 \tag{5.36}$$

where u_i is the Darcy velocity and $f(q)$ is the flux given by (2.56). As described above, we discretize these using a cell-centered approach to give the flux $u_i f_i(q)$ which holds over the entire ith cell. This nonlinear function of the solution q is then used to obtain the Riemann problem

$$q_t + F_{i-1/2}(q)_x = 0 \tag{5.37}$$

where $F_{i-1/2}$ is as given by (5.27) with $A = u$ (so that F is now a scalar) and initial data as in (5.28). For the moment, we assume that $u_i f_i'(q)$ has the same sign everywhere, i.e. over all cells, with, say $u_i f_i'(q) > 0$ so that shocks and other information will move right. Then, the Riemann solution for (5.37) is [1]

$$q(x,t) = \begin{cases} Q_{i-1} & \text{if } x < x_{i-1/2} \\ Q_{i-1}^r & \text{if } x_{i-1/2} < x < x_{i-1/2} + s_{i-1/2}^1 t \\ Q_i & \text{if } x > x_{i-1/2} + s_{i-1/2}^1 t \end{cases} \tag{5.38}$$

where

$$u_{i-1}f_{i-1}(Q_{i-1}) = u_i f_i(Q^r_{i-1/2}) \tag{5.39}$$

gives the value $Q^r_{i-1/2}$ of q just to the right of $x_{i-1/2}$ as illustrated in left hand plot of figure 5.3. There is a single propagating wave $W^1_{i-1/2}$ with speed $s^1_{i-1/2}$ given by the Rankine-

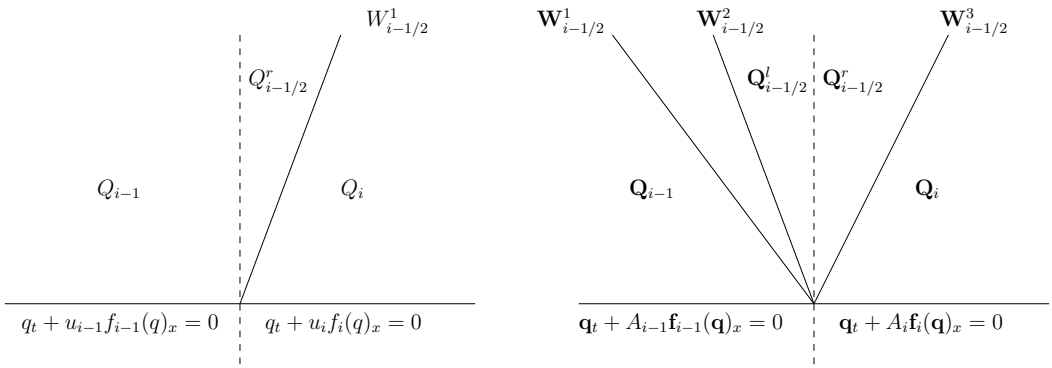


Figure 5.3: Riemann solution for the variable-coefficient equation in the case of $u_{i-1}J_{i-1} > 0$ and $u_i J_i > 0$ (left). Structure of the Riemann solution for a generalized Riemann problem with $m = 3$ (right).

Hugoniot condition,

$$s^1_{i-1/2} = \frac{u_i f_i(Q_i) - u_{i-1}f_{i-1}(Q^r_{i-1})}{Q_i - Q^r_{i-1}} \ . \tag{5.40}$$

Note that there is also a stationary discontinuity in q at $x_{i-1/2}$ that arises from the jump in the Jacobian $J_{i-1/2} = uf'$ at this point, which leads to a corresponding jump in the saturation. The flux, however, should be continuous at this point since the saturation leaving cell $i-1$ must enter cell i, and this leads to the expression (5.39). Note that this is also a special case of the Rankine-Hugoniot jump condition across the stationary discontinuity at $x_{i-1/2}$.

For equation (5.37), we can choose $r^1_{i-1/2} = 1$ as the "eigenvector" of $J_{i-1/2}$ with eigenvalue $s^1_{i-1/2}$, so that $J_{i-1/2} = s^1_{i-1/2}$. Attempting to solve the Riemann problem by a decomposition of the form (5.30) would fail here, as it would lead to $W^1_{i-1/2} = \alpha^1_{i-1/2} = Q_i - Q_{i-1}$, which is not correct. The problem is that we have neglected to take account of the jump in q at $x_{i-1/2}$. The crucial observation is that if we instead decompose the flux-difference, as opposed to the solution itself, and write the solution in terms of these flux-waves, we are able to solve this Riemann problem correctly. The update formula then uses flux-waves, as opposed to waves generated using the discontinuity in the solution, to update the cell-

averages on either side. Using the same eigenvectors of the approximate Jacobian $J_{i-1/2}$ the flux-difference is decomposed as

$$u_i f_i(Q_i) - u_{i-1} f_{i-1}(Q_{i-1}) = \sum_{p=1}^{m} \beta^p_{i-1/2} r^p_{i-1/2} \equiv \sum_{p=1}^{m} Z^p_{i-1/2} \tag{5.41}$$

where

$$\beta_{i-1/2} = R^{-1}_{i-1/2} \left(u_i f_i(Q_i) - u_{i-1} f_{i-1}(Q_{i-1}) \right) \tag{5.42}$$

(note that $m = 1$ and $R = 1$ here). This is correct since the entire flux difference is carried by the one propagating wave, with no flux difference remaining at $x_{i-1/2}$. To see that this agrees with the solution (5.38), we can obtain the correct wave $W^1_{i-1/2}$ of figure 5.3 by dividing $Z^1_{i-1/2}$ by the wave speed (as suggested by (5.41) and (5.30))

$$\begin{aligned} W^1_{i-1/2} &= Z^1_{i-1/2} / s^1_{i-1/2} & (5.43) \\ &= Q_i - Q^r_{i-1/2} & (5.44) \end{aligned}$$

using (5.40). Whilst this is useful for interpretation, we do not need to use $W^1_{i-1/2}$ since the flux waves $Z^p_{i-1/2}$ can be used directly in the wave-propagation algorithm. Using the previous notation, the the fluctuations (5.33), (5.32) used in the update formula (5.34) are now defined as

$$\begin{aligned} A^+ \Delta Q_{i-1/2} &= \sum_{p: s^p_{i-1} < 0} Z^p_{i-1/2} \ , & (5.45) \\ A^- \Delta Q_{i+1/2} &= \sum_{p: s^p_{i+1} > 0} Z^p_{i+1/2} \ , & (5.46) \end{aligned}$$

i.e. we again identify the left and right going wave contributions to the cell averages.

To summarize the above, attempting to solve the generalized Riemann problem in terms of the wave W^p as illustrated in figure 5.3 requires determining the proper jump $Q^r_{i-1/2} - Q^l_{i-1/2}$, since q is not continuous across $x_{i-1/2}$. We would need to determine waves $W^p_{i-1/2}$ that are proportional to eigenvectors $r^p_{i-1/2}$ and that also lead to states $Q^r_{i-1/2}$ and $Q^l_{i-1/2}$ satisfying (5.39). In the nonlinear case this leads to a nonlinear system of equations to solve. By working instead in terms of the flux difference, the fact the flux is continuous across $x_{i-1/2}$ works to our advantage since the entire flux difference can then be decomposed into

propagating flux-waves using the linear decomposition (5.39).

Now consider our nonlinear system of $m = 1 + K \times L$ equations (5.24),

$$\mathbf{q}_t + A(x,t)\left(\mathbf{f}(\mathbf{q})\right)_x = 0 \ . \tag{5.47}$$

A cell-centered discretization followed by a linearization leads to a Jacobian $J_i = A_i \mathbf{f}_i'$ associated with cell i. Note that $\mathbf{f}_i$ is computed numerically using (5.15) and (5.6) and (5.7) within each of coarse grid cells. Similarly, the Jacobian is computed numerically using the expressions (A.15), (A.16), (A.17), (A.18) derived in Appendix A. The terms in A are computed using (5.14). The details of computing the projected terms are given in section 5.3.4.

Suppose that this matrix J_i is nonsingular with P_i positive eigenvalues and $m - P_i$ negative eigenvalues. The generalized Riemann solution must now satisfy the equations

$$\begin{cases} \mathbf{q}_t + J_{i-1}\mathbf{q}_x = 0 & \text{if } x < x_{i-1/2} \\ \mathbf{q}_t + J_i \mathbf{q}_x = 0 & \text{if } x > x_{i-1/2} \end{cases} \tag{5.48}$$

with data (5.28). This has a bounded solution provided that $P_{i-1} = P_i \equiv P$ and that the set of m vectors obtained by taking the eigenvectors of J_{i-1} that correspond to negative eigenvalues along with the eigenvectors of J_i that correspond to positive eigenvalues forms a linearly independent set. These are the vectors we use as the $\mathbf{r}^p_{i-1/2}$ for $p = 1, 2, \ldots, m$, along with the corresponding eigenvalues as the $s^p_{i-1/2}$. Then the Riemann problem has a unique solution in the form illustrated in right hand plot of figure 5.3, with m propagating waves proportional to these vectors. The left-going $m - P$ waves satisfy the Rankine-Hugoniot conditions for the equation $\mathbf{q}_t + J_{i-1}\mathbf{q}_x = 0$ valid for $x < x_{i-1/2}$, while the right-going P waves satisfy the Rankine-Hugoniot conditions for the equation $\mathbf{q}_t + J_i \mathbf{q}_x = 0$ valid for $x > x_{i-1/2}$. At $x_{i-1/2}$ the values $\mathbf{Q}^l_{i-1/2}$ and $\mathbf{Q}^r_{i-1/2}$ must be related via $A_{i-1}\mathbf{f}_{i-1}(\mathbf{Q}^l_{i-1/2}) = A_i \mathbf{f}_i(\mathbf{Q}^r_{i-1/2})$ so that the flux is continuous. Thus, the approximate Jacobian $J_{i-1/2}$ is defined using the combination of the eigenvectors and eigenvalues from J_{i-1} and J_i as described above. The numerical scheme proceeds in exactly the way described above for the scalar case, using the expression (5.41) (with uf now replaced by $A\mathbf{f}$) to define the waves $\mathbf{Z}^p$ which are used in the update formulas.

The above methods give first order schemes (5.34), like Godunov's method, but written

in terms of the waves and the cell averages. To extend the 1-d scheme to high-resolution method, correction fluxes are added, so that (5.34) now becomes [24]

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x}\left[A^+\Delta\mathbf{Q}_{i-1/2} + A^-\Delta\mathbf{Q}_{i+1/2}\right] - \frac{\Delta t}{\Delta x}\left[\tilde{\mathbf{F}}_{i+1/2} - \tilde{\mathbf{F}}_{i-1/2}\right] \tag{5.49}$$

where

$$\tilde{\mathbf{F}}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{m}\mathrm{sgn}(s_{i-1/2}^p)\left(1 - \frac{\Delta t}{\Delta x}|s_{i-1/2}^p|\right)\tilde{\mathbf{Z}}_{i-1/2}^p\ . \tag{5.50}$$

$\tilde{\mathbf{Z}}_{i-1/2}^p$ is a "limited" version of the wave $\mathbf{Z}_{i-1/2}^p$ obtained by comparing $\mathbf{Z}_{i-1/2}^p$ to $\mathbf{Z}_{I-1/2}^p$, the corresponding wave from the adjacent Riemann problem on the upwind side, where

$$I = \begin{cases} i-1 & \text{if } s_{i-1/2}^p > 0 \\ i+1 & \text{if } s_{i-1/2}^p < 0 \end{cases} \tag{5.51}$$

If no limiter is applied, so that $\tilde{\mathbf{Z}}_{i-1/2}^p = \mathbf{Z}_{i-1/2}^p$ then for the linear problem, this method reduces to the Lax-Wendroff method. Using a limiter reduces the non-physical oscillations that are present in that method, and allows for robust and accurate computations of shocks and other discontinuous solutions.

To use these schemes for multidimensional methods, we can either use dimension splitting, as mentioned above, or directly apply the wave-propagation methodology. In this second case, the method is developed in much the same way as was done above, where we solve approximate Riemann problems at each of the cell edges individually and then use the flux-continuity over cell edges to determine the flux across each edge and hence the contribution to the cell averages. These methods, and their extension to higher order schemes are well described in [24].

As mentioned, we use the hyperbolic package CLAWPACK to solve the homogeneous system (5.17). Some further details of doing this are given in Appendix C.

5.3.3 Computation of Source Terms

As mentioned in the introduction of this section, we use a fractional step method to handle the source terms present in our upscaled equations (5.12). Thus, to update the contribution

from the source terms, we use equation (5.18),

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{H}(\mathbf{q}, \boldsymbol{x}, t) \tag{5.52}$$

where $\mathbf{H}$ is the vector function

$$\mathbf{H}(\mathbf{q}, \boldsymbol{x}, t) = (0, G_{11}, G_{12}, \ldots, G_{KL})^T \tag{5.53}$$

with the function G given by

$$G(\mathbf{q}, \boldsymbol{x}, \boldsymbol{y}, t) = \left(\overline{\boldsymbol{v}} + \mathcal{P}(\boldsymbol{v}')\right) \cdot \mathcal{Q}\left(\nabla_x f'\right) - \mathcal{P}\left(\mathcal{Q}(\boldsymbol{v}') \cdot \mathcal{Q}\left(\nabla_x f'\right)\right) + \mathcal{Q}\left(\frac{\partial S'}{\partial t}\right) \ . \tag{5.54}$$

The benefit of using this fractional step method is that we can take advantage of the schemes that are available for the hyperbolic part, described above, and also (5.18) becomes a set of ODEs in $\mathbf{q}$ and hence we can take advantage of the schemes available for solving these. The type of splitting that we used here is known as a Godunov splitting and is generally only first-order accurate. A modification to the method yields a formally second order scheme, known as Strang's method. However, the difference in accuracy in the solution is minimal for the coarse-grids that we are attempting to use here and does not justify the additional cost (c.f. examples in [24]).

The general implementation of this method is very simple, the only issue being the choice of ODE solver to be used. A wide variety of ODE solvers are available for systems of the form $y' = \psi(y, t)$ where $y(t) \in \mathbb{R}^n$. Note, however, that in general we cannot use multi-step methods that require more than one level of data (e.g. y^{n-1} as well as y^n) to generate the solution y^{n+1} at the next time level. This is because we only have data $\mathbf{Q}_i^*$ (the result of solving the hyperbolic part at this stage) to use in computing $\mathbf{Q}_i^{n+1}$. Previous values (e.g. $\mathbf{Q}_i^n$ or $\mathbf{Q}_i^*$ from the previous time step) are not suitable to use in the context of multi-step methods because $\mathbf{Q}_i^*$ is computed from $\mathbf{Q}_i^n$ by solving a different equation (the hyperbolic equation (5.17)) than the ODE (5.18) we are now attempting to approximate. Because of this issue, we use Runge-Kutta methods to solve (5.18) which do not use previous values of

the solution. The following classical second-order, two-stage method is used

$$\mathbf{Q}_i^{**} = \mathbf{Q}_i^* + \frac{\Delta t}{2}\mathbf{H}(\mathbf{Q}_i^*, \boldsymbol{x}, t^n) \tag{5.55}$$

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^* + \frac{\Delta t}{2}\mathbf{H}(\mathbf{Q}_i^{**}, \boldsymbol{x}, t^n)\ . \tag{5.56}$$

In numerical tests, this scheme was found to be stable with the same time step Δt that was used for the updating of the hyperbolic part (5.17) via the wave-propagation methods.

The other issue that remains are the actual details of the computation for the terms in $\mathbf{H}$, i.e. $\nabla_x \cdot \overline{\boldsymbol{v}'f'}$ and the computation of G. For $\nabla_x \cdot \overline{\boldsymbol{v}'f'}$ we first compute the average $\overline{\boldsymbol{v}'f'}$ with f' computed using the solution $\mathbf{Q}_i^*$ from the hyperbolic part. As mentioned previously, this average is computed using the 2-d equivalent of the trapezoidal rule on the sub-grid. The Darcy velocity $\boldsymbol{v}'$ is computed at the start of the overall time step. Note that we do not update $\boldsymbol{v}'$ at the intermediate steps (e.g. using the solution $\mathbf{Q}_i^{**}$) since it depends on the solution $\mathbf{Q}$ which is not available at these intermediate points. The averaging gives a cell centered values for $\overline{\boldsymbol{v}'f'}$ (cell-centered with respect to the coarse grid). We then perform finite-differencing of these to obtain an approximation for $\nabla_x \cdot \overline{\boldsymbol{v}'f'}$. It was found that upwind differencing is needed for stability (upwind with respect to the coarse velocity $\overline{\boldsymbol{v}}$), i.e. for $\left(\overline{\boldsymbol{v}'f'}\right)_{x_1}$, if $\overline{\boldsymbol{v}}_1 > 0$ then we use the approximation

$$\left(\overline{\boldsymbol{v}'f'}\right)_{x_1} \approx \frac{\left(\overline{\boldsymbol{v}'f'}\right)_{i,j} - \left(\overline{\boldsymbol{v}'f'}\right)_{i-1,j}}{\Delta x_1} \tag{5.57}$$

otherwise if $\overline{\boldsymbol{v}}_1 < 0$ then we use

$$\left(\overline{\boldsymbol{v}'f'}\right)_{x_1} \approx \frac{\left(\overline{\boldsymbol{v}'f'}\right)_{i+1,j} - \left(\overline{\boldsymbol{v}'f'}\right)_{i,j}}{\Delta x_1}\ . \tag{5.58}$$

The x_2-derivatives are approximated in a corresponding manner. Despite being first order approximations, these were found to be adequate. When second order central difference approximations were used, spurious oscillations formed in the solution. $\nabla_x \cdot \overline{\boldsymbol{v}'f'}$ is recomputed at the intermediate time using $\mathbf{Q}_i^{**}$ in exactly the same way.

The computation of the function G is more involved since the terms all involve using the streamline average $\mathcal{P}$ or the related projection $\mathcal{Q}$. For the moment, assume that the velocity, its average and fluctuation have been computed. The streamline projection of the

velocity fluctuations $\mathcal{P}(v_1)$, $\mathcal{P}(v_2)$ will already have been computed from the velocity solver and the preparation stages for the hyperbolic solver. We give the details of the method we use to compute the streamline projection numerically in the next section. $\mathcal{Q}(\boldsymbol{v}')$ is also easily computed from $\mathcal{Q}(\boldsymbol{v}') = \boldsymbol{v}' - \mathcal{P}(\boldsymbol{v}')$. The remaining terms to be computed therefore are $\mathcal{Q}(\nabla_x f')$ and $\mathcal{Q}\left(\frac{\partial S'}{\partial t}\right)$. In the case where we prescribe a form for the velocity field, as in the testing sections that immediately follow this, we can compute these using Lemma 4.13. In this case, we do not actually need to compute the derivatives of S' nor f'. The computation of these projections then involves simply computing derivatives of $\boldsymbol{v}$ (see the derivation of Lemma 4.13 and the details for the streamline projection given below). By employing the multiscale finite element method, described in section 5.4.2, we are also able to compute the velocity derivatives needed in Lemma 4.13 in the more practical case where the permeability field is prescribed.

5.3.4 Computation of Streamline Projection

From the section on the multiscale analysis, we see that the streamline projection $\mathcal{P}$ is a fundamental component of the upscaling scheme since it eliminates the fast-time dependence. In numerically computing this quantity we have the option of using the two different forms: via (4.48) and Corollary 4.1, or else via (4.122) and (4.123). Whilst the second of these is useful for interpretation, it was found that using this form to numerically compute the projection was cumbersome. Thus, to compute the projections $\mathcal{P}$ and $\mathcal{Q}$ we use (4.48) and Corollary 4.1. Recall that in order to obtain $\mathcal{Q}(u)$ we must solve

$$\nabla_y \cdot (A \nabla_y \theta) = \boldsymbol{v} \cdot \nabla_y u \tag{5.59}$$

with periodic boundary conditions where A is the 2×2 matrix with components $A_{ij} = v_i v_j$. Then, $\mathcal{Q}(u) = \boldsymbol{v} \cdot \nabla_y \theta$. $\mathcal{P}(u)$ is then obtained via $\mathcal{P}(u) = u - \mathcal{Q}(u)$. The equation (5.59) is degenerate because A is singular, having eigenvalues 0 and $v_1^2 + v_2^2$. Because of this, solving this equation is more difficult than a standard elliptic equation, where most solution methods rely on A be positive-definite. For example, we had no success in using finite-element methods to try to solve (5.59) numerically. As noted in Chapter 4, the solution of (5.59) is unique only up to a function in $\mathcal{N}$.

Because of the difficulties associated with A being singular, to solve (5.59) we instead

consider the related equation

$$\frac{\partial g}{\partial \mu} = \nabla_y \cdot (A \nabla_y g) - \boldsymbol{v} \cdot \nabla_y v \tag{5.60}$$

with periodic boundary conditions, where μ is an "artifical time". The steady state solution of this equation is clearly a solution of (5.59). Thus, the idea is to solve (5.60) to a steady state using time stepping, starting from an initial guess of the solution. To do this we employ a semi-implicit discretization,

$$\frac{g^{n+1} - g^n}{\Delta \mu} = \alpha \nabla^2 g^{n+1} + \nabla_y \cdot (A \nabla_y g^n) - \alpha \nabla^2 g^n - \boldsymbol{v} \cdot \nabla_y u \tag{5.61}$$

where α is a constant, chosen to improve the rate of convergence to the steady state. Rearranging terms in this equation gives

$$\left(1 - \alpha \Delta \mu \nabla^2\right) g^{n+1} = \left(1 - \alpha \Delta \mu \nabla^2\right) g^n + \Delta \mu \nabla_y \cdot (A \nabla_y g^n) - \Delta \mu \boldsymbol{v} \cdot \nabla_y u \tag{5.62}$$

so that at each time step, we solve a constant coefficient elliptic equation for g^{n+1}. This is easily accomplished on a uniform Cartesian grid and fast-Fourier transforms (FFT). The grid corresponds exactly to the subgrid $\boldsymbol{y}_{kl}$ described above. In addition, all derivatives were calculated using the FFT. This method of solution is similar to that used by Ceniceros and Hou [7] and Oberman [25].

Using (5.62) with $\Delta t = 1/K$ we marched to a steady state, which was when $\| g^{n+1} - g^n \|_{l_2}$ was less than a specified tolerance (usually 10^{-6}). The initial guess was taken simply to be $g = 0$ everywhere (note that it would have been better to use the g computed from the previous time step of the hyperbolic solver, but that this would have required too much memory). The convergence rate was generally found to be rather slow, particularly for velocity fields $\boldsymbol{v}$ with complicated features. Several experiments were done to try to determine an optimal value for α that would give both a decent convergence rate and also be robust enough so that the scheme converged over a wide range of velocity fields. The value of α that seemed to work best was

$$\alpha \approx 0.7 \max_Y \sqrt{v_1^2 + v_2^2} \ , \tag{5.63}$$

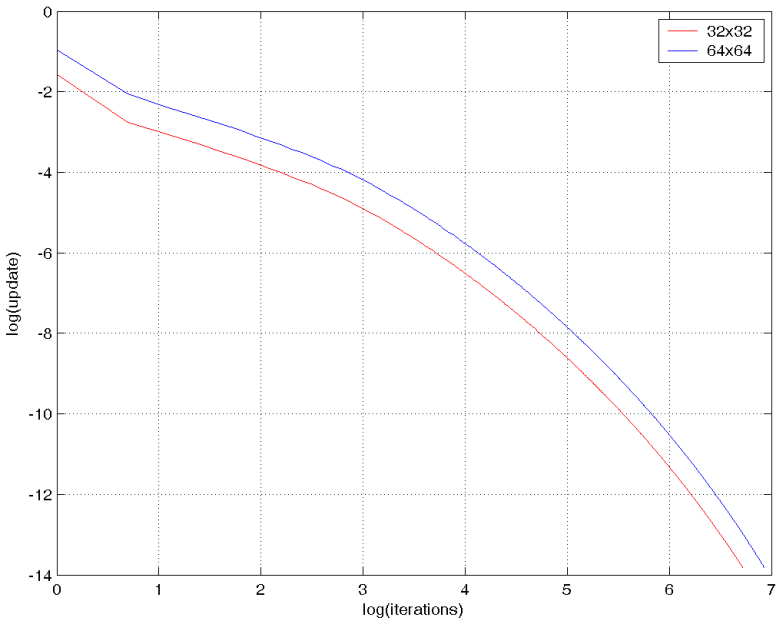


Figure 5.4: Log-log convergence plot of l_2-norm of the update in the projection computations, using 32×32 and 64×64 grids.

with the maximum taken over the sub-grid in each coarse cell (i.e. a different α is used in each coarse grid cell). Figure 5.4 shows the log-log plot of the convergence rate of the scheme for the particular velocity field given by (5.64) and (5.65) with $x_1 = 0.5$, $x_2 = 0.5$ and when taking the projection of the v_1' component of it. These are computed using 32×32 and 64×64 points for the sub-grid. Figure 5.5 shows v_1'. It takes 827 and 1022 steps to achieve the desired tolerance respectively on these grids. The important property of the projection is that it returns a function $p \in \mathcal{N}$, i.e. with $\boldsymbol{v} \cdot \nabla_y p = 0$. Figure 5.6 shows $p = \mathcal{P}(v_1')$ and figure 5.7 shows $\boldsymbol{v} \cdot \nabla_y p$ for the case of a 32×32 sub-grid. We see that to within a reasonable degree, our computed p lies in $\mathcal{N}$. The l_2-norm of this error is 0.001839.

5.3.5 Consistency and Convergence of our Scheme

It is important that we evaluate each part of the numerical scheme separately to ensure that it behaves as our analysis predicts. Therefore, since we have completely described the numerical implementation of the hyperbolic part of the scheme, we present some results

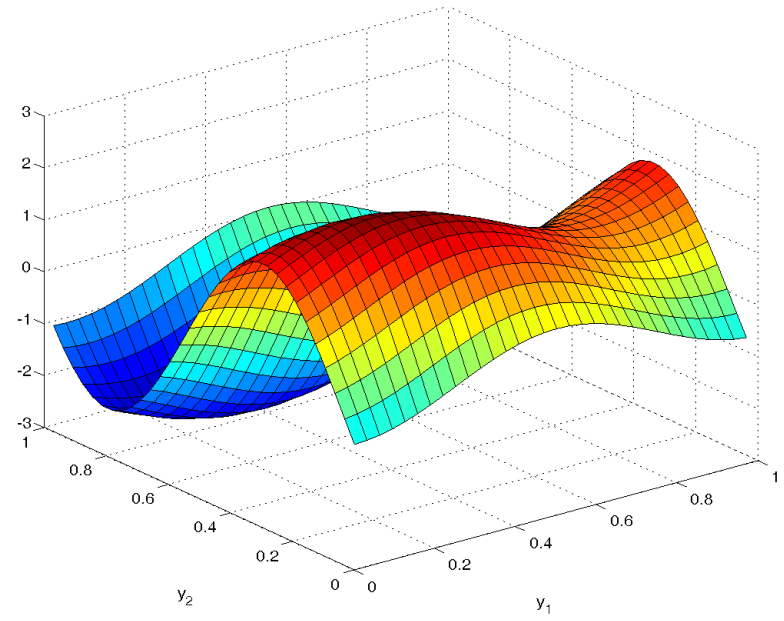


Figure 5.5: Plot of v_1' in a single cell.

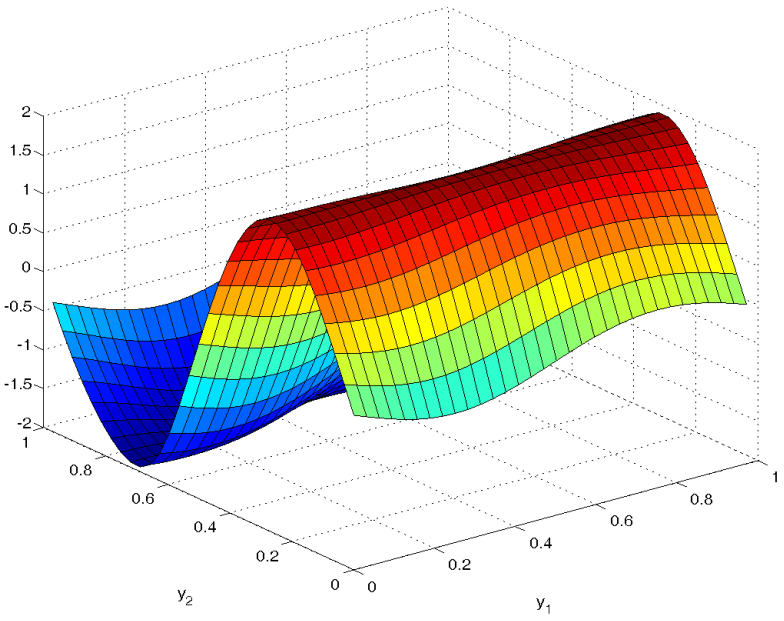


Figure 5.6: Plot of $\mathcal{P}(v_1')$ in a single cell.

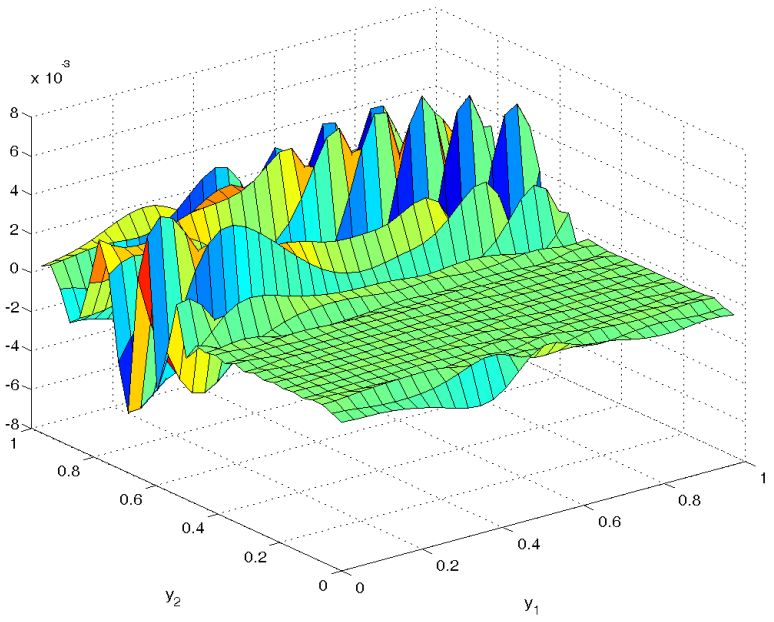


Figure 5.7: Plot of $\boldsymbol{v} \cdot \nabla_y p$, for $p = \mathcal{P}(v_1')$ in a single cell.

that demonstrate the method works. We compare the results for our upscaling scheme with resolved computations, computing error norms.

To avoid the issues involved in computing the velocity field numerically, we prescribe an analytical form for the velocity. We choose one such that the divergence-free property holds and also has fluctuations with a periodic structure. Any such velocity should be "realistic" in that it should mimic features that are typically seen in porous media flows. For instance, the fluctuations should be $O(1)$ and have a sufficiently complicated structure so that the streamline projection is not trivial to compute e.g. shear flows. The following velocity field provides a reasonable model upon which to test the scheme:

$$\overline{\boldsymbol{v}}(\boldsymbol{x}) = (4 + \cos(6\pi x_2), 0) \tag{5.64}$$

$$\boldsymbol{v}'(\boldsymbol{x}, \boldsymbol{y}) = \frac{2}{3}\pi \sin(4(x_1 + x_2)) \cos(2\pi(y_1 + y_2))(1, -1) + (\sin(2\pi y_1), 0)\ . \tag{5.65}$$

The streamlines of this flow are shown in figure 5.8. The streamlines are close to that for a shear flow, but one can see the oscillations in the vertical directions. Note that this velocity field does not satisfy the no-flow boundary conditions on the top and bottom edges and

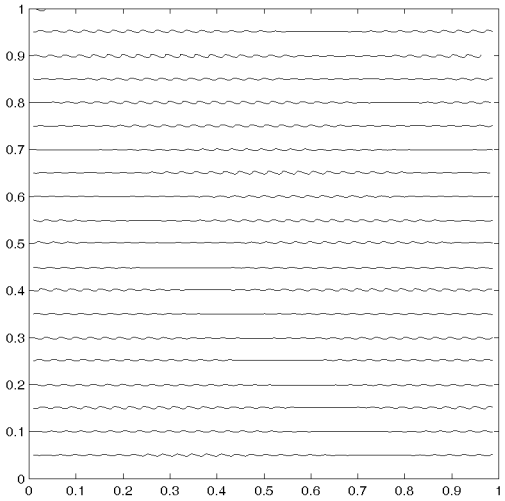


Figure 5.8: Streamlines for the velocity field used in testing the scheme for the saturation equation.

so the model does not correspond exactly to that for the core-plug model that we wish to replicate. To be consistent, however, we choose periodic boundary conditions on these edges for the hyperbolic problem. Since the flow is mainly in the positive x_1-direction this is only a minor issue. Finally, we need to prescribe an initial saturation for the problem. For this we use an initially smooth (i.e. with no spatial fluctuations) function S_0 given by:

$$S_0 = \begin{cases} 1 & x_1 < \left(b - \frac{\delta}{4}\right) \\ \frac{1}{2}\left[1 - \sin\left(\frac{2\pi(x_1 - b)}{\delta}\right)\right] & |x_1 - b| < \frac{\delta}{4} \\ 0 & x_1 > \left(b - \frac{\delta}{4}\right) \end{cases} \tag{5.66}$$

This initial saturation is a smoother version of the initial profile that we used earlier in the resolved computations for the full flow problem, as in it is an jump centered at b mollified by the parameter δ. The "density plot" of the initial saturation S_0 is shown in figure 5.9 for the particular values $\delta = 0.8$ and $b = 0.3$. Bright red corresponds to all water regions and dark blue corresponds all oil regions.

We perform a convergence analysis for the scheme using the setup described above. All computations are done on our usual unit domain. Although the velocity field is given analytically, an analytical form for the evolution saturation is hard to determine. Therefore,

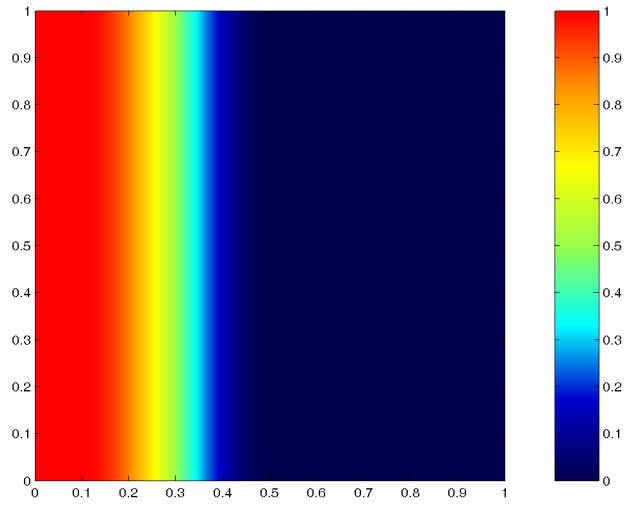


Figure 5.9: Density plot for the initial saturation.

as is often done in the evaluation of numerical schemes for homogenization, we take well resolved computations to be our "exact" solution. The scheme is tested with different coarse-grids, which for simplicity will always be uniform with an equal number of grid blocks in the x_1- and x_2-directions. We keep the number of grid points for the cell problems alway the same. By analysis similar to that described in the evaluation of the streamline projection, we determined that 32×32 grid points were sufficient to accurately compute the fluctuations.

We compare the results for the both the homogenized solution, i.e. $\overline{S}$ and also the $\overline{S}+S'$, which by our multiscale analysis, should give the exact solution S_ϵ to within $O(\epsilon)$. Note that since we have prescribed the form of the velocity field analytically, the value of the small-scale parameter ϵ does not actually appear anywhere in the computations (all the evaluations for the cell problem are scaled so that ϵ does not appear explicitly). However, in comparing with the exact solution we must choose a particular value for ϵ. We choose $\epsilon = 1/128$ which is much smaller that the mesh size of the largest coarse-grid. We choose this value since it allows us to more easily reconstruct the solution $\overline{S} + S'$, which we will refer to as the "multiscale reconstruction". We do not, however, take any advantage of the fact that this ϵ is rational.

We compute solutions on the coarse-grids $N \times N$, with $N = 16, 32, 64$. The "exact"

solution is computed on a 2048×2048 grid. For the resolved computations, we compute a homogenized solution by taking the numerical spatial average (5.6). For both $\overline{S}$ and $\overline{S} + S'$ we compute discrete error norms. We compute the l_2-norm and the infinity norm, which are given respectively by

$$\| U - U_h \|_{l^2} = \left(\sum \int_k (U - U_h)^2 d\boldsymbol{x} \right)^{\frac{1}{2}} \tag{5.67}$$

$$\| U - U_h \|_{\infty} = \max_k |U - U_h| \tag{5.68}$$

where k is a grid block and K is the set of grid blocks over the domain. Here we take U to be the resolved saturation (or average) and U_h the corresponding saturation computed via our upscaling scheme. For the "exact" average saturation we compute the average of the resolved solution using a numerical quadrature, with the area of integration corresponding to exactly one oscillation cell. This was done at points corresponding to the centers of the coarse grid used in the upscaled calculations.

We first show the results for the more simple case of single-phase flow ($f(S) = S$ in the saturation equation). Figure 5.10 shows the resulting solution at time $t = 0.1$. This time is sufficient to allow the fluctuations in the saturation to form whilst maintaining the whole front to remain within the domain. From the figure we can see the saturation "fingers" that develop. Figure 5.11 shows the corresponding solutions computed from our upscaling scheme. Comparing with the resolved computations we see that the scheme accurately captures the overall profile and the fluctuations. Indeed, the results on the 64×64 coarse-grid appear almost identical.

Tables 5.2 and 5.1 show the errors in the homogenized solution and the multiscale reconstruction for the single-phase case. These are also illustrated in on the log-log plots in figures 5.14 and 5.13. By computing the slope of the line of best-fit on these plots we get the order of accuracy of the scheme. We see that the scheme is first-order accurate as expected since the updating of the source terms is first-order.

If we ignore the interaction terms in the average equation (5.1) then the average saturation is not captured correctly. To see this, we show the average saturation contours for the upscaling method above and in the case when the interaction term $\nabla \cdot \overline{\boldsymbol{v}' f'}$ is ignored. These are shown in figures 5.15 for exactly the same setup as the previous plots. Comparing the plots, which are shown with grid lines for ease of comparsion, we see that if the interaction

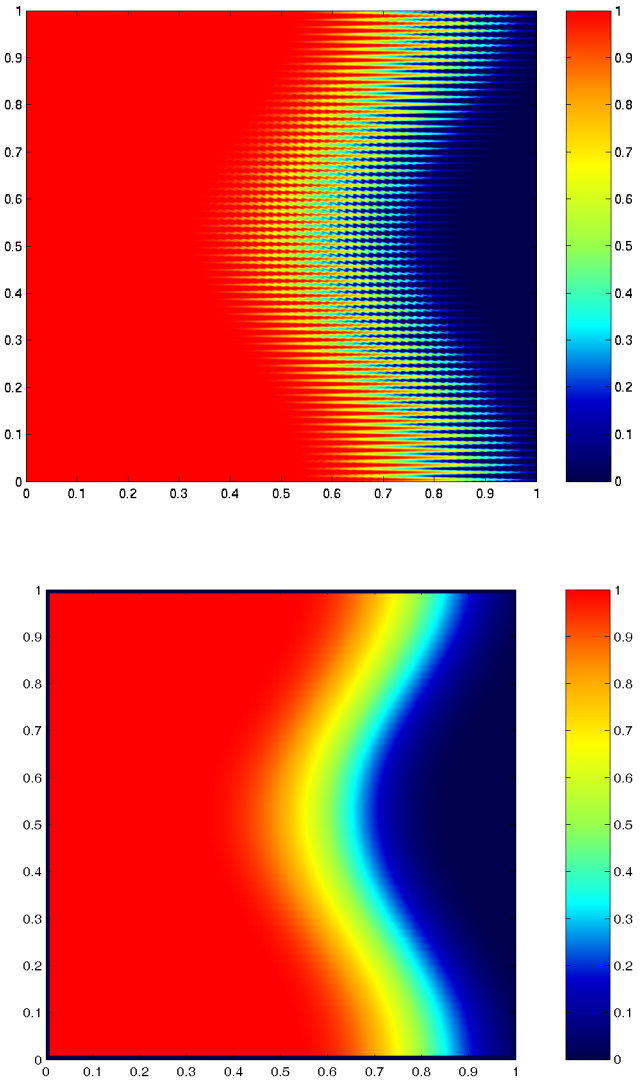


Figure 5.10: Density plot for the "exact" solution for the saturation in the single-phase case, resolved solution and average.

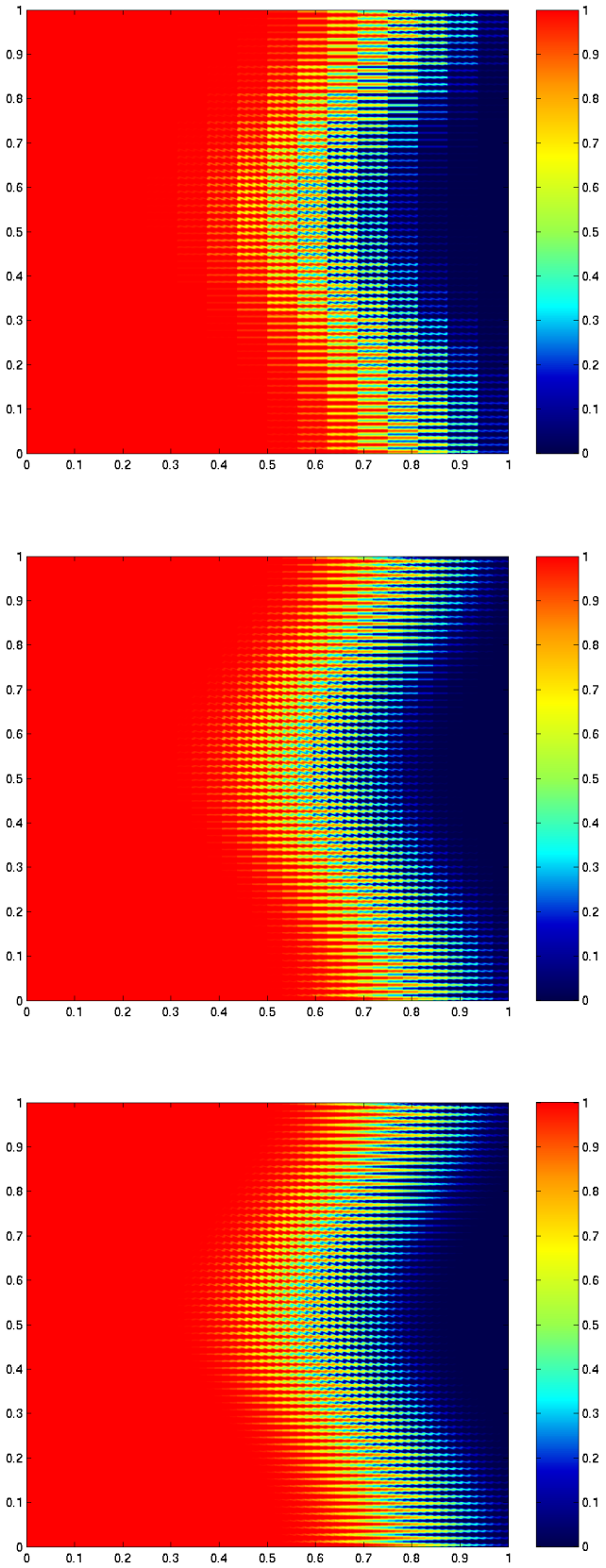


Figure 5.11: Multiscale reconstruction of the saturation from the upscaling scheme for the single-phase case using 16×16, 32×32 and 64×64 coarse-grids.

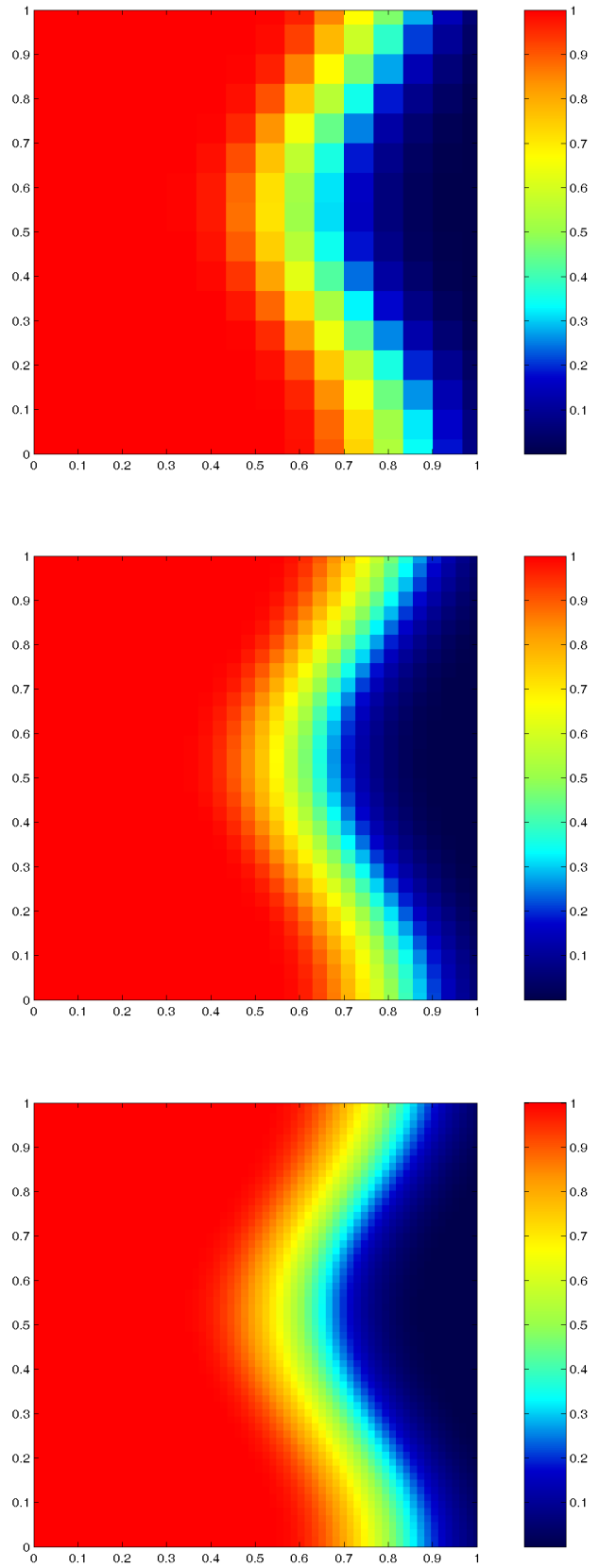


Figure 5.12: Average saturation from the upscaling scheme for the single-phase case using 16×16, 32×32 and 64×64 coarse-grids.

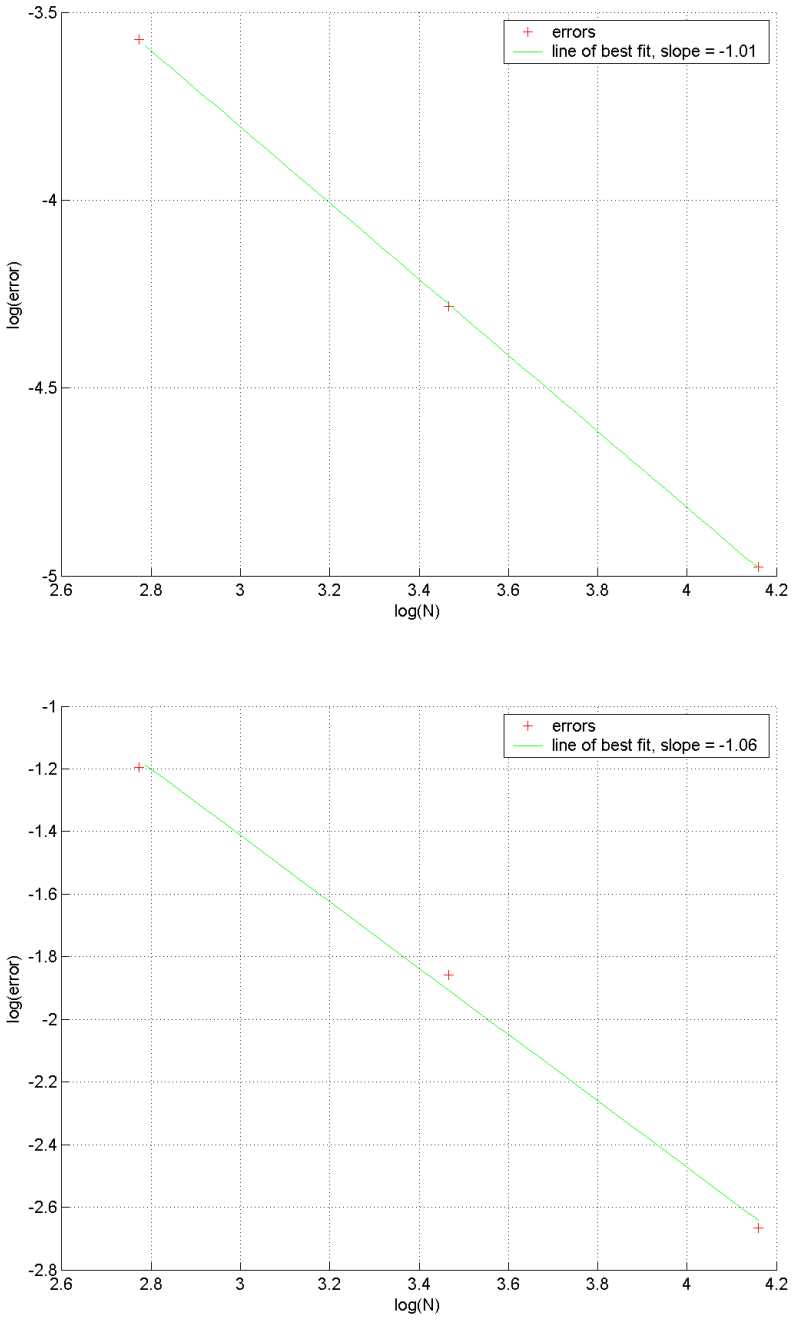


Figure 5.13: Log-log error plots for the l^2 and max norm error respectively in the multiscale reconstructed solution for the single-phase case, demonstrating first-order convergence.

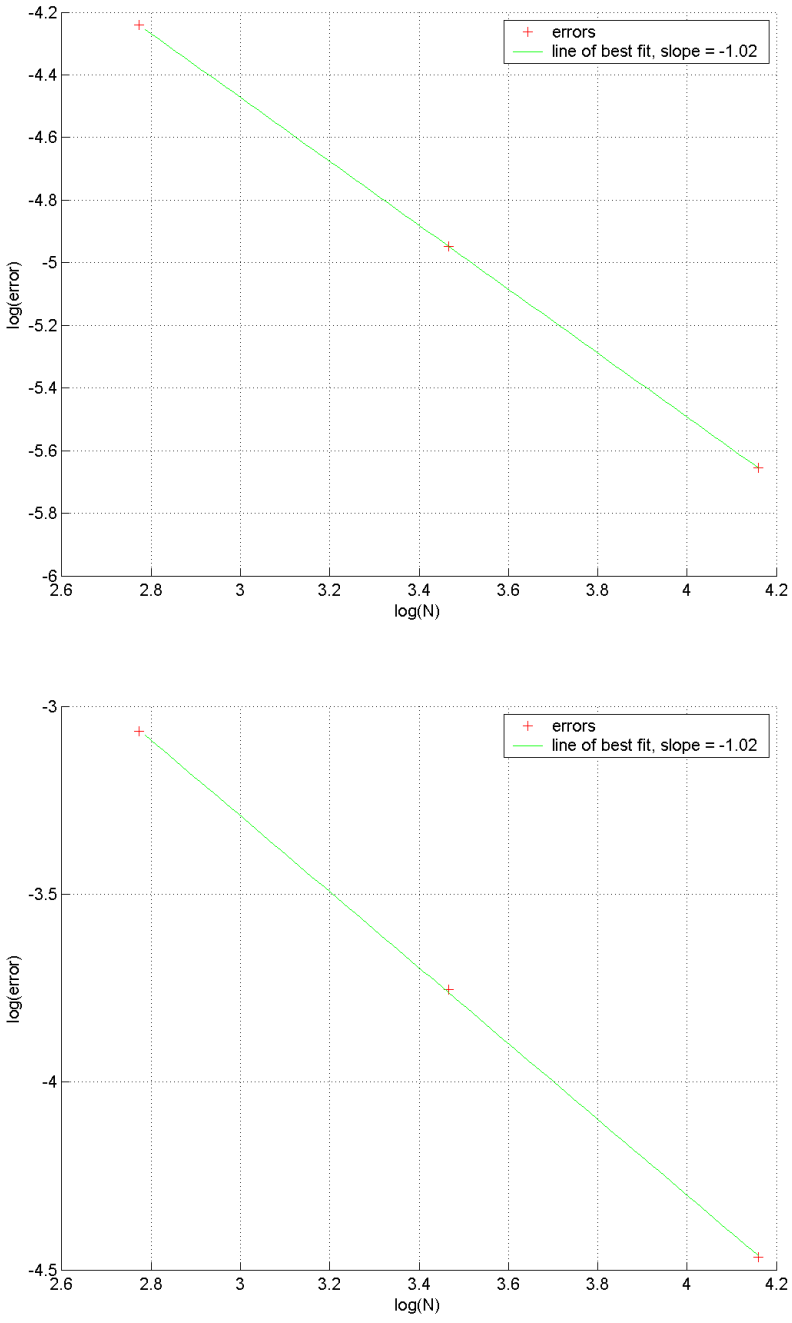


Figure 5.14: Log-log error plots for the l^2 and max norm error respectively in the homogenized solution for the single-phase case, demonstrating first-order convergence.

N	l^2 error	max error
16	0.0276	0.3012
32	0.0140	0.1541
64	0.0069	0.0686

Table 5.1: Errors in the multiscale reconstruction in the single-phase case using different grids.

N	l^2 error	max error
16	0.0144	0.0460
32	0.0070	0.0233
64	0.0034	0.0071

Table 5.2: Errors in the homogenized solution in the single-phase case using different grids.

terms are ignored, we do not get the right amount of "spreading" of the average saturation front. Since the time at which the comparison is done is not very great, the difference is only small. However, even at this stage it is clearly noticable and in longer simulations this would grow.

We next show the results for the two-phase flows. We again use the same velocity field and same initial data for the tests. In addition, we use exactly the same grids for both the coarse and sub-grids. Tables 5.4 and 5.3 show the errors in the solution. Figure 5.16 shows the resolved calculations, figure 5.17 shows the results of using the upscaled calculations. We again see that the upscaled calculations capture the solution well.

In this case, we see that the error convergence is less than first-order, being approximately 0.4 for the reconstructed solution and approximately 0.7 for the homogenized solution. This is due to the fact that a shock has already formed in the solution by this time and so the formal convergence rates no longer apply. To check that the scheme has first-order rate of convergence, we computed the solution before the shock forms, at $t = 0.05$. The corresponding errors are shown in Tables 5.6 and 5.5 show clearly that the scheme is first-order.

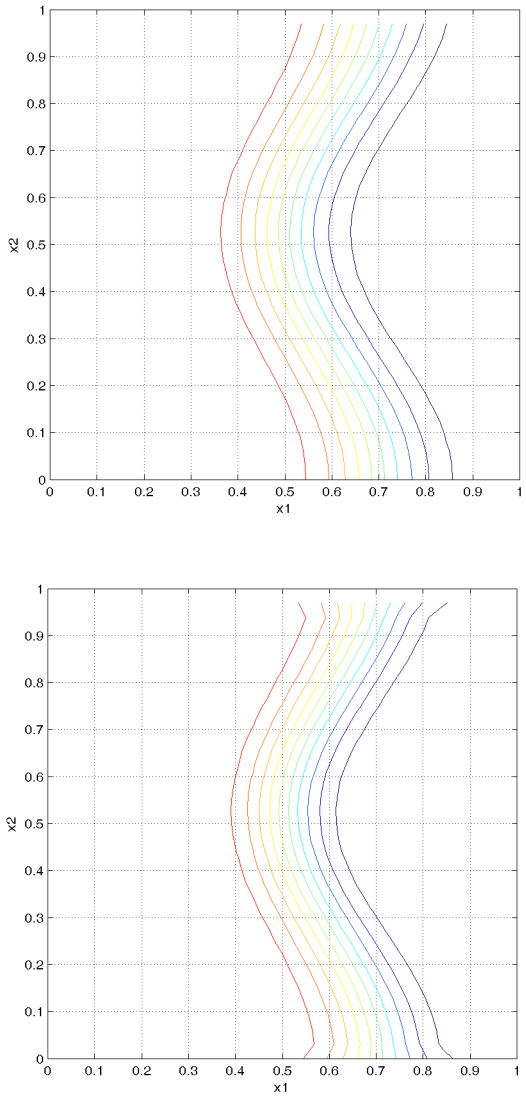


Figure 5.15: Comparison of the saturation contours for the upscaling scheme (top) and in the case where the interaction terms are ignored (bottom).

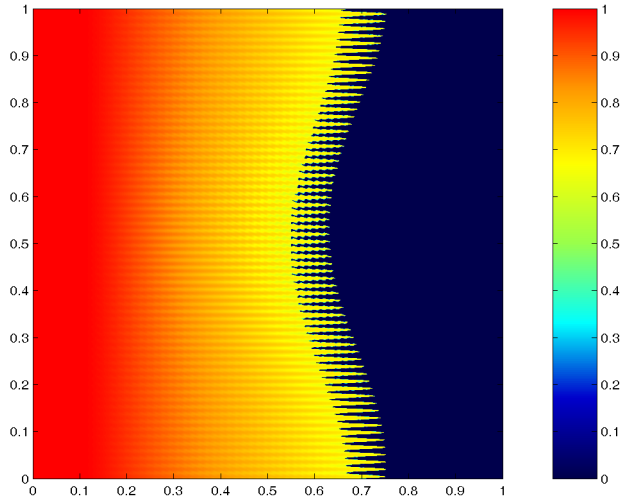


Figure 5.16: Density plot for the "exact" solution for the saturation.

N	l^2 error	max error
16	0.0697	0.7007
32	0.0524	0.6959
64	0.0467	0.6811

Table 5.3: Errors in the multiscale reconstruction in the two-phase case using different grids.

N	l^2 error	max error
16	0.0723	0.0354
32	0.0433	0.0251
64	0.0275	0.0212

Table 5.4: Errors in the homogenized solution in the two-phase case using different grids.

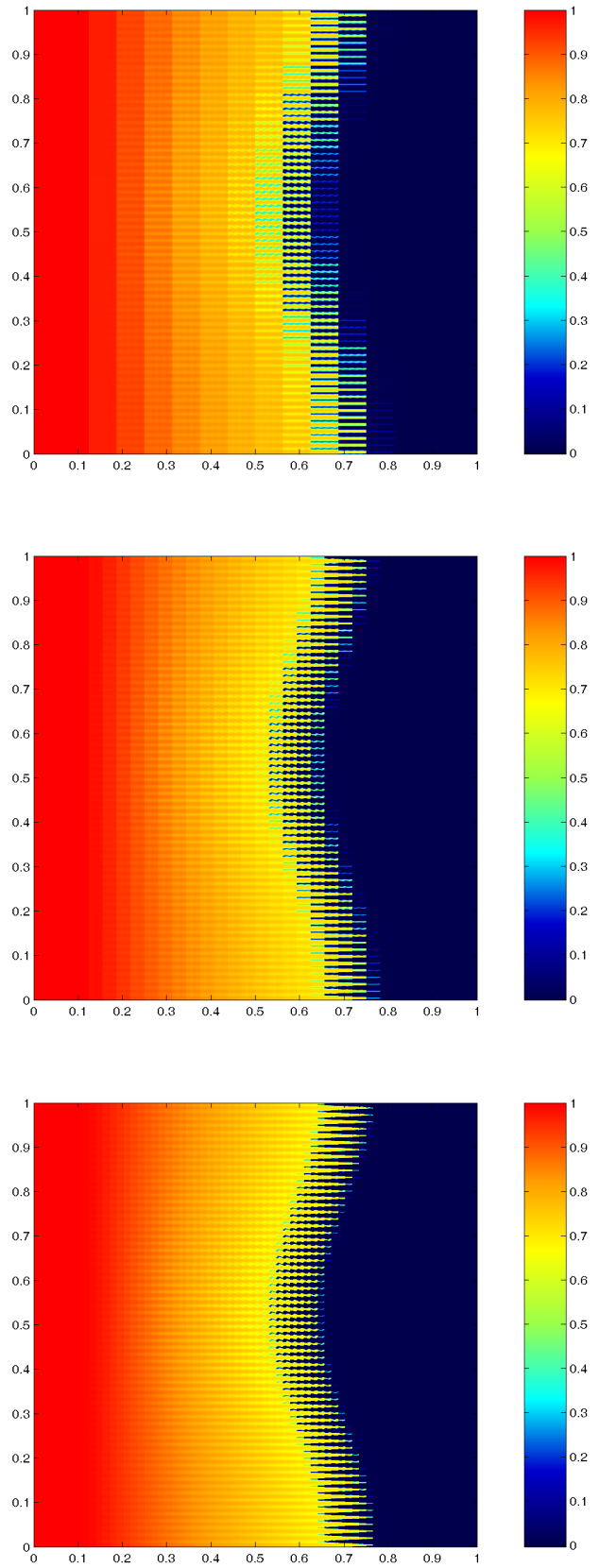


Figure 5.17: Multiscale reconstruction of the saturation from the upscaling scheme for the two-phase case using 16×16, 32×32 and 64×64 coarse-grids.

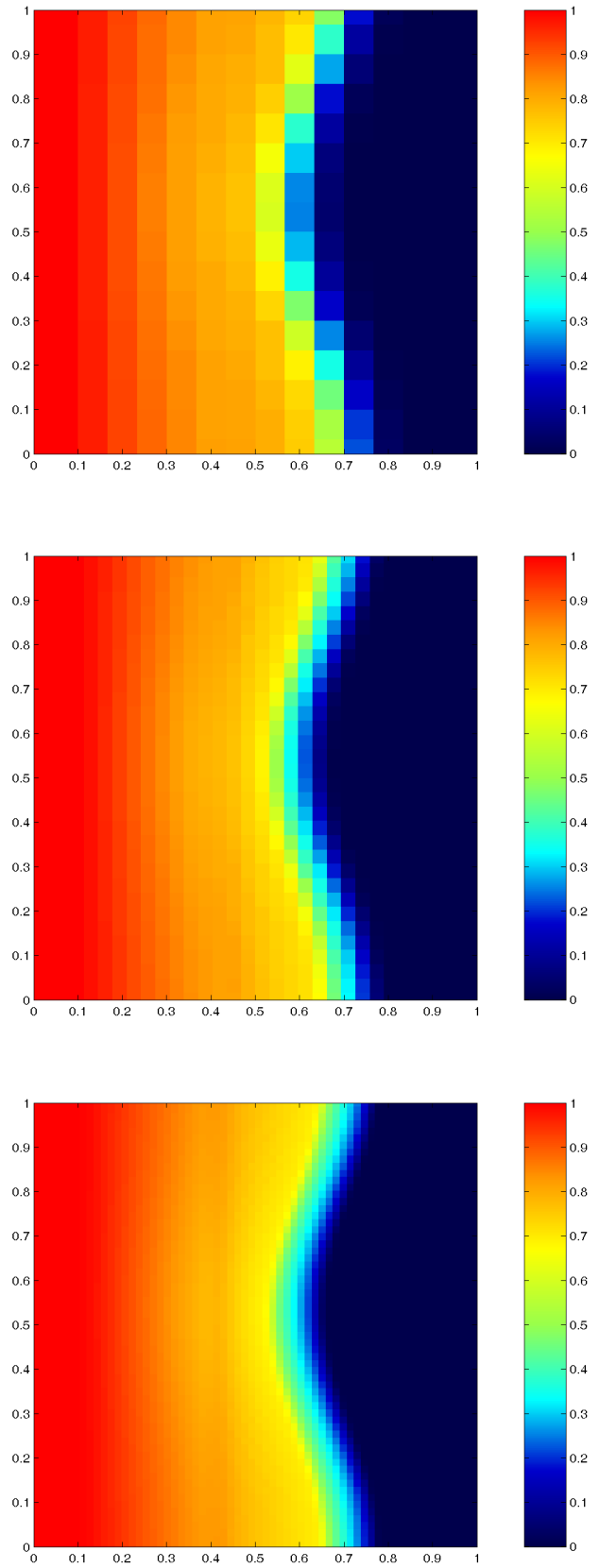


Figure 5.18: Average saturation from the upscaling scheme for the two-phase case using 16×16, 32×32 and 64×64 coarse-grids.

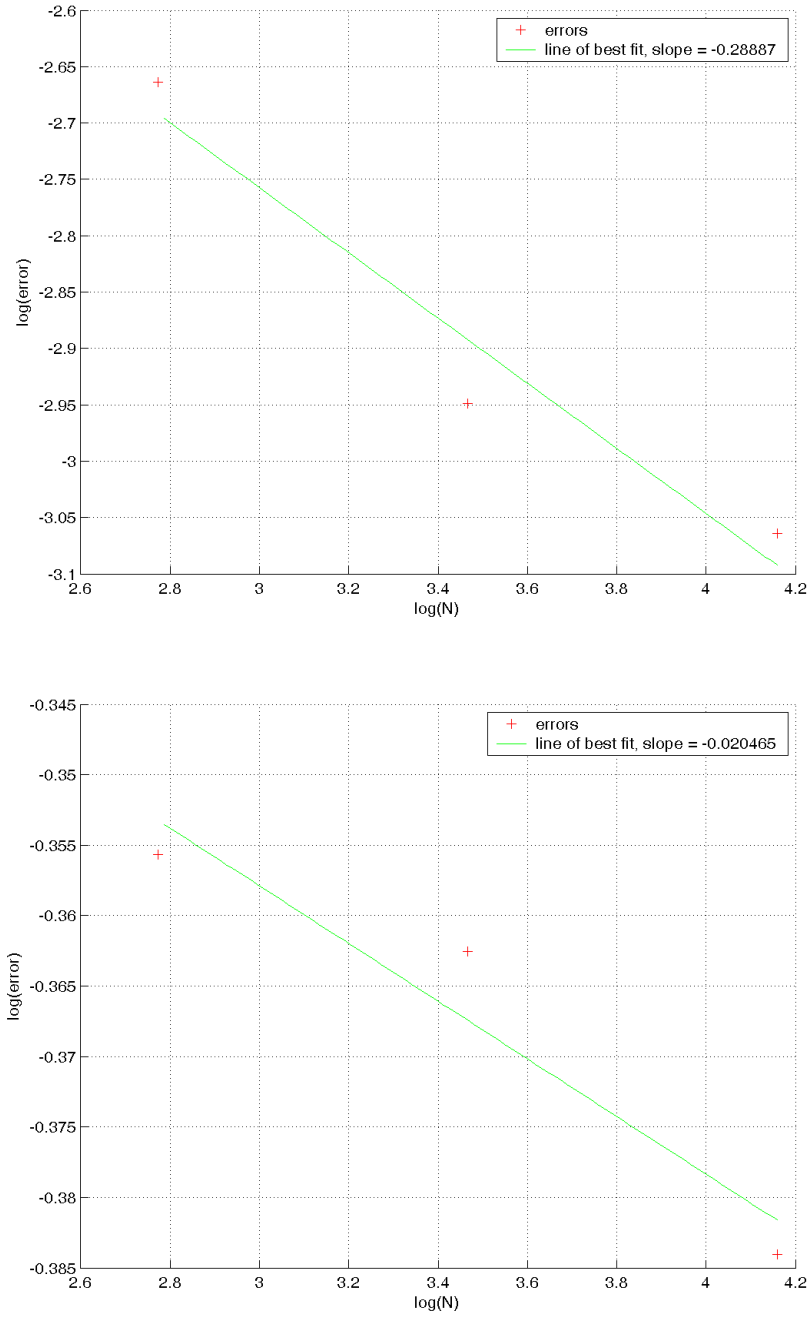


Figure 5.19: Log-log error plots for the l^2 and max norm error respectively in the multiscale reconstruction for the single-phase case.

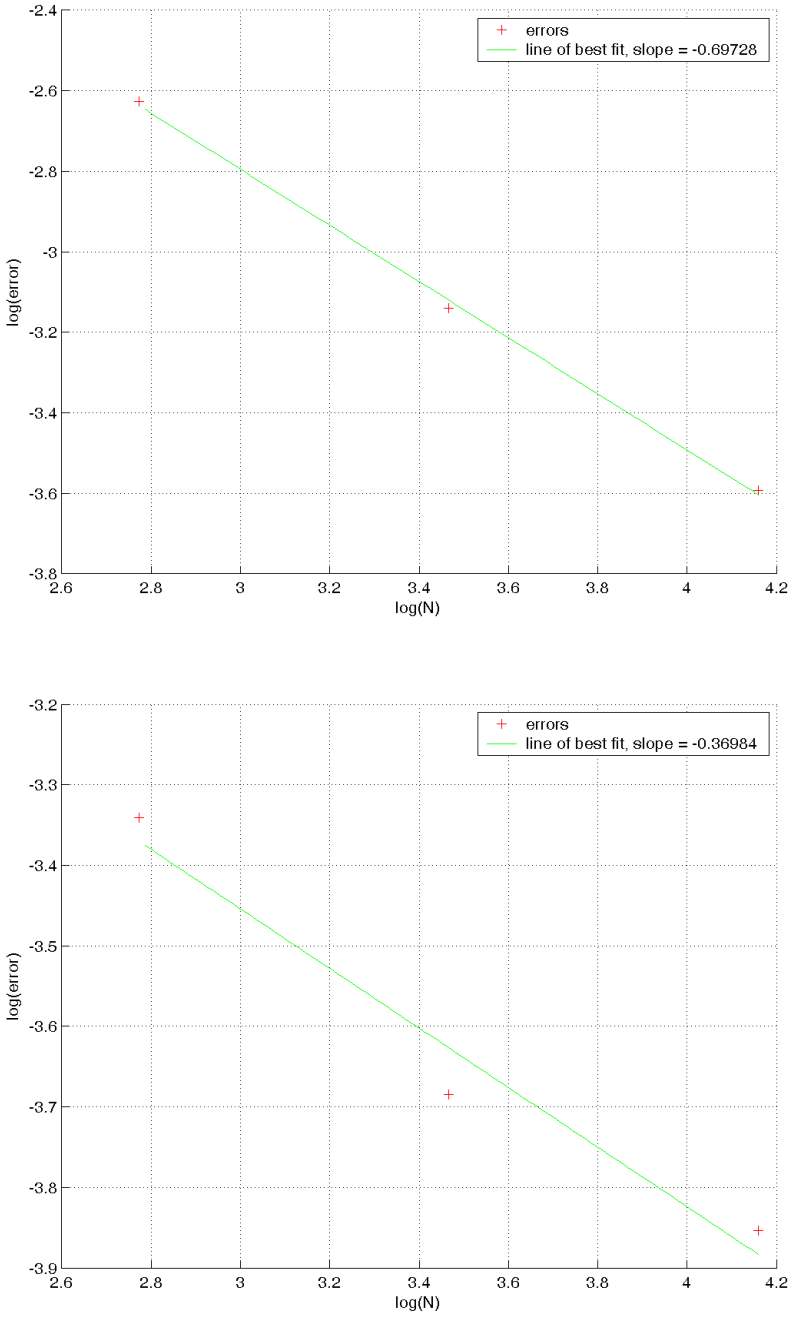


Figure 5.20: Log-log error plots for the l^2 and max norm error respectively in the homogenized solution for the single-phase case.

N	l^2 error	max error
16	0.0184	0.1120
32	0.0095	0.0764
64	0.0049	0.0461

Table 5.5: Errors in the multiscale reconstruction in the two-phase case using different grids, before shock forms.

N	l^2 error	max error
16	0.0270	0.0831
32	0.0148	0.0495
64	0.0076	0.0264

Table 5.6: Errors in the homogenized solution in the two-phase case using different grids, before shock forms.

5.4 Numerical Upscaling Method for the Pressure/Velocity Equation

We now turn to the problem of developing a numerical upscaling scheme for the pressure/velocity equation (PVE) (2.51). The multiscale analysis of section 4.3 shows us that for permeability fields with $O(1)$ oscillations (in the fast variable $\boldsymbol{y}$) we can expect that the velocity field (needed in the saturation equation) will also have oscillations which are $O(1)$.

The goal of the upscaling scheme is of course to be able to solve (2.51) on a coarse-grid. Moreover, infact, we wish to use the same coarse-grid that is used for the saturation scheme. Should we wish to get only the solution at coarse-grid points, i.e. only the average $\overline{\boldsymbol{v}}$, then we would look to solve the homogenized equation (4.24) with either an approximation for the "equivalent" permeability $a^*(\boldsymbol{x}, t)$ or else use expression (4.25) which is valid in the case of periodic oscillations. As mentioned in Chapter 1, there have been several attempts to calculate equivalent permeabilities for different flow situations. However, in our scheme for the saturation equation, we see that we need to have not only the average velocity but also the fluctuations $\boldsymbol{v}'$, e.g. in equation (5.1) where we need to evaluate the interaction term $\nabla_x \cdot \overline{\boldsymbol{v}' f'}$. Therefore, the scheme must be capable of providing both. At first this may appear a contradictory goal: we wish to upscale, i.e. solve the equation on a coarse-grid, and yet get be able to get fine scale information within the same scheme. This contradiction can be resolved if we realize that may be able to somehow *interpolate* the coarse-grid solution, using only locally computed quantities, to get fine scale information within the interior of

coarse-grid cells. We are fortunate that such a method with this philosophy already exist and incorporating it into our scheme is just a matter of effective implementation. This method is the multiscale finite element. The method is special in the sense that it can be viewed as an upscaling method, and yet they also provide a handle to fine scale information. This latter feature is sometimes referred to as *downscaling*.

We give a description of these methods in the sections that follow.

5.4.1 Multiscale Finite Element Method

The multiscale finite element (MSFEM) for elliptic problems is fully described in [19], [20], [16]. We will give an outline of the method and then describe the adaptations needed to use it efficiently in our scheme.

Consider our elliptic pressure equation (2.51)

$$-\nabla \cdot (a(\boldsymbol{x}, t)\nabla p) = f \tag{5.69}$$

where, as in the section on multiscale analysis, we have written $a = \lambda(S)K$. Since a depends on the saturation it is actually time dependent. However, since we are advancing the numerical solution to (2.51), (2.52) by alternately solving each seperately, we can consider t as a parameter when solving the pressure equation. In our case, the source term f will be zero everywhere unless there exist source or sinks within the domain. However, for generality in describing the method we leave it in (5.69). For the moment, $\boldsymbol{x}$ is used to denote a general spatial variable, rather than the coarse-grid variable used above.

For the purpose of deriving the method, we first suppose that (5.69) holds in a domain Ω and that $p = 0$ on $\partial\Omega$. The modifications for handling inhomogeneous boundary conditions are trivial. The variational problem of (5.69) is then to seek $p \in H_0^1(\Omega)$ such that

$$a(p, q) = f(q), \quad \forall q \in H_0^1(\Omega) \tag{5.70}$$

where

$$a(p, q) = \int_\Omega a(\boldsymbol{x}) \frac{\partial q}{\partial x_i} \frac{\partial p}{\partial x_i} d\boldsymbol{x} \tag{5.71}$$

$$f(q) = \int_\Omega f(\boldsymbol{x}) q(\boldsymbol{x}) d\boldsymbol{x} \tag{5.72}$$

(summation convention is used for repeated indices here). Since $a(\boldsymbol{x}, t)$ is bounded from below by a positive constant, the linear form $a(\cdot, \cdot)$ is elliptic and continuous, i.e.

$$\alpha \left|q\right|_{1,\Omega}^2 \leq a(q,q), \quad \forall q \in H_0^1(\Omega) \tag{5.73}$$

and

$$\left|a(p,q)\right| \leq \beta \left|p\right|_{1,\Omega} \left|q\right|_{1,\Omega}, \quad \forall q \in H_0^1(\Omega) \tag{5.74}$$

A *finite element* method is obtained by restricting the weak formulation (5.70) to a finite dimensional subspace of $H_0^1(\Omega)$. Let $\mathcal{K}^h$ be a partition of Ω of elements K with diameter less than h. In our method we will always assume that the partition consists of rectangular elements which are defined by an axi-parallel rectangular mesh and with maximum edge length h. This case covers the meshes we described in the previous sections for the saturation equation. Let $\boldsymbol{x}_s \in \overline{K}$ $(s = 1, \ldots, d)$, $d = 4$, be the nodal points of K. In each element $K \in \mathcal{K}^h$, we define a set of *basis functions* $\{\phi_K^r, i = 1, \ldots, d\}$. In the traditional finite element method these basis functions would be bilinear function [5]. In MSFEM, these basis functions satisfy

$$-\nabla \cdot (a(\boldsymbol{x}) \nabla \phi_K^r) = 0 \tag{5.75}$$

inside K. As is usual for finite element basis functions we require $\phi_K^r(\boldsymbol{x}_s) = \delta_{rs}$. Further, we need to specify the boundary conditions of ϕ_K^r for well posedness of (5.75). The choice of boundary conditions on the basis functions has a strong influence on the convergence of MSFEM. For the moment we assume that the boundary conditions are linear along the boundaries of the elements, i.e. along the boundaries MSFEM basis functions and traditional (linear) finite element basis functions coincide.

MSFEM with these basis functions is conforming, i.e.

$$V^h = \operatorname{span}\{\phi_K^r : i = 1, \ldots, d; K \in \mathcal{K}^h\} \subset H_0^1(\Omega) \tag{5.76}$$

and the approximate solution of (5.69) in V^h, i.e. $p^h \in V^h$ is

$$a(p^h, q) = f(q), \quad \forall q \in H_0^1(\Omega)\ . \tag{5.77}$$

We can see that the only difference between MSFEM and a traditional finite element method

is in the construction of the basis functions. Note that (5.77) is solved at the coarse grid level, with the solution p^h given at coarse grid nodes, and it is in this sense that it is an upscaling scheme.

The purpose of introducing MSFEM is for upscaling the pressure equation in the case where two length scales exist. In this case we are assuming, as in the multiscale analysis of Chapter 4, that $a = a(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x}$ is slow variable and $\boldsymbol{y} = \boldsymbol{x}/\epsilon$ is the fast variable. For the linear finite element method, convergence is possible only by using meshes with element size $h < \epsilon$. When ϵ is small this is clearly a restrictive condition. Note that in this case, MSFEM is very similar to the linear finite element method since within each element K, a will have little variation and hence (5.75) with linear boundary conditions will give basis functions ϕ_K^r that are close to bilinear functions. The great benefit of MSFEM is that convergence is possible for $h \gg \epsilon$. In this case, the basis functions will be quite different from the linear case. In general the ϕ_K^r will have a oscillatory component (i.e. that depends on the fast scale $\boldsymbol{y}$) within K. Because the basis functions satisfy the same equation as the pressure equation (5.69) this oscillatory component captures the small scale features in the same way as the exact pressure p.

The statements above are made precise in the paper of Hou, Wu and Cai [20] for the case when a is a period function of the fast-variable $\boldsymbol{y}$. In the case where $h \ll \epsilon$, the following estimates for the convergence of the solution p^h are derived

$$\| p - p^h \|_{1,\Omega} \leq C \left(\frac{h}{\epsilon} \right) \| f \|_{1,\Omega} \tag{5.78}$$

and

$$\| p - p^h \|_{0,\Omega} \leq C \left(\frac{h}{\epsilon} \right)^2 \| f \|_{0,\Omega} \ . \tag{5.79}$$

Here C is used to denote a generic constant. This result holds for both the linear finite element method and also MSFEM. However, for MSFEM we also have the following estimates valid for $h \gg \epsilon$,

$$\| p - p^h \|_{1,\Omega} \leq C_1 h \| f \|_{0,\Omega} + C_2 \left(\frac{\epsilon}{h} \right)^{\frac{1}{2}} \tag{5.80}$$

and

$$\| p - p^h \|_{0,\Omega} \leq C_1 h^2 \| f \|_{0,\Omega} + C_2 \epsilon + C_3 \frac{h}{\epsilon} \ . \tag{5.81}$$

Again C_1 and C_2 are used to denote generic constants. Note that the approximation p^h is

interpolated over the basis functions, i.e. for p^h in K

$$p^h(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} p^h(\boldsymbol{x}_i)\phi_K^r(\boldsymbol{x}, \boldsymbol{y}) \tag{5.82}$$

where $\boldsymbol{x}_i$ are the nodal points of K. Thus in the case where the fast oscillations are resolved by the coarse grid, MSFEM has similar convergence properties to (linear) FEM. However, in the case where the coarse grid does not resolve the fast oscillations we still get convergence for MSFEM (in this case, the features in the basis functions resolve the fast oscillations). Generalizations of the results (5.80) and (5.80) to cases where the oscillations are non-periodic can be found in [16].

As mentioned above, the choice of boundary conditions imposed on the basis functions can have a significant influence on the quality of the resulting solution p^h. This was investigated in detail in [19] and [20]. Imposing linear boundary conditions gives rise to a boundary layer near ∂K in the solution for the basis functions. To overcome this problem in choosing the boundary conditions, an oversampling scheme for the construction of the basis functions was proposed in [19]. The basis functions in this oversampling method are constructed in the following way. We first construct the basis functions ψ^ρ in a sampling domain $S \supset K$ (see figure 5.21) by solving

$$\nabla \cdot (a(\boldsymbol{x})\nabla\psi^\rho) = 0 \tag{5.83}$$

where the ψ^ρ are piecewise linear along ∂S and $\psi^\rho(\boldsymbol{x}_\sigma) = \delta_{\rho\sigma}$ at the nodal points of S. For simplicity we assume that S is rectangular and hence having the same number of nodal points as K. Moreover, we choose S sufficiently large so that $\text{diam}(S) = h_1 > h$ and ∂S is away from ∂K at a distance of order ϵ. Next, the basis functions ϕ^r on K are constructed from the linear superposition of ψ^ρ

$$\phi_r^k = \sum_{\rho=1}^{d} c_{r\rho}^k \psi^\rho \tag{5.84}$$

where r is the index of the nodal point and constants $c_{r\rho}^k$ are determined by the condition $\phi_r^k(\boldsymbol{x}_s) = \delta_{rs}$, $\boldsymbol{x}_s$ being the nodal points of K. By this procedure, the boundary layer structure near ∂S is avoided.

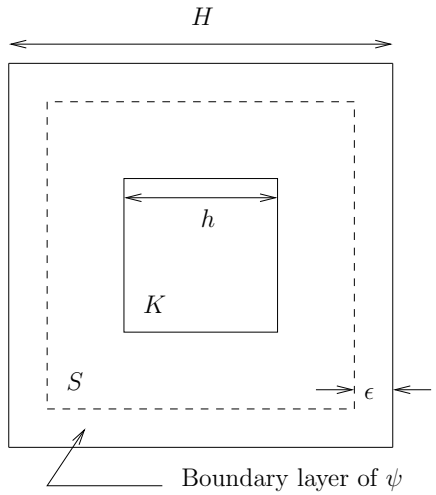


Figure 5.21: Oversampled basis function construction, using a samples from a larger domain to avoid the boundary effect.

An important consequence of the above construction is that the basis functions ϕ^r are no longer continuous across the internal boundaries of the elements, nor are they zero on the external boundaries. Setting the basis functions to be zero outside the external boundaries of the elements introduces first order discontinuities of ϕ along all sides of the elements ([16]). Thus, these basis functions are non-conforming and V^h spanned by ϕ is no longer in H^1. This complicates the analysis (the details of which can be found in [16]). The improved convergence results for $h \gg \epsilon$ are now

$$\| p - p^h \|_{1,\Omega} \leq C_1 h + C_2 \left(\frac{\epsilon}{h} \right) + C_3 \sqrt{\epsilon} \tag{5.85}$$

and

$$\| p - p^h \|_{0,\Omega} \leq C_1 h^2 + C_2 \epsilon + C_3 \left(\frac{\epsilon}{h} \right)^2 + C_4 \epsilon \left| \ln h \right| \ . \tag{5.86}$$

Whilst the above results were derived under the assumption of periodic oscillations in a, numerical tests show that the method works well in cases where this may not hold.

5.4.2 Special MSFEM for the Case of Periodic Oscillations

In the case of periodic oscillations such as those that have been considered in Chapter 4 we may employ a variant of the MSFEM that gives a numerical two-scale solution directly, i.e. a solution of the form $p = p(\boldsymbol{x}, \boldsymbol{y})$. Recall that the basis functions in MSFEM satisfy

(5.75) within a coarse grid cell. a is assumed to be of the form $a = a(\boldsymbol{x}, \boldsymbol{y})$ and periodic in $\boldsymbol{y}$. In exactly the same way as the pressure equation had an asymptotic expansion of the form (4.14), the basis functions will have an expansion of the form

$$\phi_K^r = \phi_{0,K}^r(\boldsymbol{x}) + \epsilon\phi_{1,K}^r(\boldsymbol{x}, \boldsymbol{y}) + \epsilon^2\phi_{2,K}^r(\boldsymbol{x}, \boldsymbol{y}) + O(\epsilon^3) \ . \tag{5.87}$$

Using an exactly similar analysis that was used to derive the expressions (4.24) and (4.21) for p_0 and p_1 and now applied to (5.87) we get

$$\frac{\partial}{\partial x_i}\left(a^*(\boldsymbol{x})\frac{\partial \phi_{0,K}^r}{\partial x_i}\right) = 0 \tag{5.88}$$

and

$$\phi_{1,K}^r(\boldsymbol{x}, \boldsymbol{y}) = -\chi^j \frac{\partial \phi_{0,K}^r}{\partial x_j} \ . \tag{5.89}$$

where a^* is a diagonal tensor with

$$a_{ij}^*(\boldsymbol{x}) = \frac{1}{|Y|}\int_Y a(\boldsymbol{x}, \boldsymbol{y})\left(1 - \frac{\partial \chi^j}{\partial y_i}\right) d\boldsymbol{y} \tag{5.90}$$

and χ^j satisfies

$$\frac{\partial}{\partial y_i}\left(a(\boldsymbol{x}, \boldsymbol{y})\frac{\partial \chi^j}{\partial x_i}\right) = \frac{\partial a}{\partial y_j} \tag{5.91}$$

with periodic boundary conditions.

The coarse grid will always have sufficient resolution so that the elements in $a^*(\boldsymbol{x})$ can be approximated by constants throughout a coarse grid cell K. Thus, (5.88) with linear boundary conditions will have solutions that correspond to the standard bilinear basis functions. That is, the $\phi_{0,K}^r$ are the standard bilinear basis functions. Then, we take

$$\begin{aligned} \phi_K^r &= \phi_{0,K}^r + \epsilon\phi_{1,K}^r & (5.92)\\ &= \phi_{0,K}^r - \epsilon\chi^j \frac{\partial \phi_{0,K}^r}{\partial x_j} & (5.93) \end{aligned}$$

which gives the basis functions to within $O(\epsilon^2)$. To determine the basis functions in all the coarse grid blocks, we need to solve (5.91) within each to obtain χ^j, $j = 1, 2$ and then use these in (5.93).

Once the solution for the pressure has been obtained at coarse grid points, via the

usual construction of the stiffness matrix and solving the resulting linear equations we can reconstruct the two-scale pressure within each cell using

$$p = \sum_r^d p_r^h \phi_K^r \tag{5.94}$$

$$= \sum_r^d p_r^h \left(\phi_{0,K}^r - \epsilon \chi^j \frac{\partial \phi_{0,K}^r}{\partial y_j} \right) + O(\epsilon^2) \ . \tag{5.95}$$

We call this a two-scale numerical solution since it has variation with respect to the coarse grid (each p_r^h is given at coarse grid node) and variation within the cell (χ^j is varying as a function of $\boldsymbol{y}$ inside the cell). We can prove the convergence of this MSFEM, and this is given in Appendix B along with numerical examples. To obtain the two-scale velocity field for use in our scheme for the saturation equation, we use Darcy's law

$$\boldsymbol{v} = -a(\boldsymbol{x}, \boldsymbol{y}) \nabla p \tag{5.96}$$

$$= -a(\boldsymbol{x}, \boldsymbol{y}) \left(\frac{\partial}{\partial x_i} + \frac{1}{\epsilon} \frac{\partial}{\partial y_i} \right) \left[\sum_r^d p_r^h \left(\phi_{0,K}^r - \epsilon \chi^j \frac{\partial \phi_{0,K}^r}{\partial x_j} \right) \right] + O(\epsilon) \tag{5.97}$$

$$= -a(\boldsymbol{x}, \boldsymbol{y}) \sum_r^d p_r^h \left(\frac{\partial \phi_{0,K}^r}{\partial x_i} - \frac{\partial \chi^j}{\partial y_i} \frac{\partial \phi_{0,K}^r}{\partial x_j} \right) + O(\epsilon) \ . \tag{5.98}$$

We compute $\boldsymbol{v}$ at the center of each coarse grid cell using (5.98), and then take the average of this to obtain $\overline{\boldsymbol{v}}$ and $\boldsymbol{v}'$ which are then used in the scheme for the saturation equation. Note that χ^j and hence $\boldsymbol{v}'$ are solved for on exactly the sub-grid described in the previous sections for the saturation equation, and the average velocity is computed as a cell-centered quantity on the coarse grid. This is consistent with our hyperbolic scheme described in section 5.3.2. Further, from the above expression we can compute $\boldsymbol{v}_{x_i}$ which is needed for computing $\mathcal{Q}(\nabla_x f')$ via Lemma 4.13. From the expression for $\boldsymbol{v}$ above we get

$$\frac{\partial v_i}{\partial x_1} = -\frac{\partial a}{\partial x_1} \sum_r^d p_r^h \left(\frac{\partial \phi_{0,K}^r}{\partial x_i} - \frac{\partial \chi^j}{\partial y_i} \frac{\partial \phi_{0,K}^r}{\partial x_j} \right)$$
$$-a(\boldsymbol{x}, \boldsymbol{y}) \sum_r^d p_r^h \left(\frac{\partial^2 \phi_{0,K}^r}{\partial x_1 \partial x_i} - \frac{\partial^2 \chi^j}{\partial x_1 \partial y_i} \frac{\partial \phi_{0,K}^r}{\partial x_j} - \frac{\partial \chi^j}{\partial y_i} \frac{\partial^2 \phi_{0,K}^r}{\partial x_1 \partial x_j} \right) \tag{5.99}$$

and similarly for the x_2-derivatives. In order to compute the above, we need the x-derivatives of χ^j. To derive an expression for these, take the x-derivative of equation (5.91) (which

defines χ^j). Doing this and rearranging the result gives:

$$\frac{\partial}{\partial y_i}\left[a(\boldsymbol{x},\boldsymbol{y})\frac{\partial}{\partial y_i}\left(\frac{\partial \chi^j}{\partial x_1}\right)\right] = \frac{\partial^2 a}{\partial x_1 \partial y_j} - \frac{\partial}{\partial y_i}\left(\frac{\partial a}{\partial x_1}\frac{\partial \chi^j}{\partial y_i}\right) \ , \tag{5.100}$$

i.e. an elliptic equation for $\frac{\partial \chi^j}{\partial x_1}$. Again, a similar result holds for the x_2-derivative. We can solve these equations very easily once we have solved for χ^j.

The only disadvantage to using a finite-element method in solving the pressure equation is that the divergence-free property of the velocity field (4.31), in particular, $\nabla_x \cdot \overline{\boldsymbol{v}} = 0$, nor $\nabla_x \cdot \boldsymbol{v}' = 0$ are not explicitly enforced (note that $\nabla_y \cdot \boldsymbol{v}' = 0$ by construction). Thus, the resulting velocity fields are not exactly divergence-free. This was generally not found to be a problem for our computations since the velocity field is used in the upscaled equations (5.1) and (5.2) which does not have a conservation property, unlike the original equation (4.32). When MSFEM is used for the velocity field in (4.32) it was found that it gave poor results for long time simulations [8, 21]. To overcome these problems, a mixed multiscale finite element method was introduced in [8], and in [21] a multiscale finite volume method was introduced. However, neither is as simple to implement as the MSFEM described above, which is why we implemented this method.

5.4.3 Implementation of MSFEM

The implementation of MSFEM is relatively straightforward, being similar to a traditional finite element method. The only differing issue is in the construction of the basis functions. For the case of periodic coefficients we use the method described above for the construction of the basis functions. This involves solving for the two functions χ^j, $j = 1, 2$ given by (5.91). To solve this equation we employ the same numerical scheme as that used for the computation of the projections $\mathcal{P}$ and $\mathcal{Q}$, i.e. a semi-implicit discretization of the form (5.61). Since $a(\boldsymbol{x},\boldsymbol{y}) > 0$ everywhere, (5.91) is uniformly elliptic, and the convergence rate of the scheme is much improved than that for the projections $\mathcal{P}$ and $\mathcal{Q}$. Once the χ^j are computed we take the derivatives using FFTs. These are then used in the construction of the stiffness matrix, which is formed from the evaluation of (5.77) over all the combinations of basis functions over the coarse grid. This gives a set of linear equations. The resulting equations are modified to incorporate the boundary conditions given in (3.3.1) and (3.3.2). Then, we solve the equations using a multigrid solver. The solver of DeZeeuw [33] was

found to be very fast and robust. Once the pressures are the coarse grid nodes have been found, we then evaluate (5.98) at each cell center and use this to obtain the cell-centered average velocity $\overline{\boldsymbol{v}}$ and velocity fluctuations $\boldsymbol{v}'$.

Once the basis functions have been solved for, the solution at the coarse-grid is extremely cheap. However, the construction of the basis functions is relatively expensive. Since we will be solving the pressure equation at each time step, it is worth discussing an efficient implementation of the method as it applies to our problem. In principal, since a depends on the saturation, we need to update the basis functions at each time step. The crucial observation is that for many regions, the saturation will be evolving slowly and therefore it is not necessary to update the basis functions in these regions. We can selectively choose which basis functions to update based upon how much the saturation within the cell has changed. An obvious region where basis functions would need to be updated often is near the oil-water front. Regions ahead of this front, where the water saturation is zero would need no updating, and regions behind, where the saturation is evolving more slowly, would need updating only at a less frequent intervals. Figure 5.22 illustrates the idea.

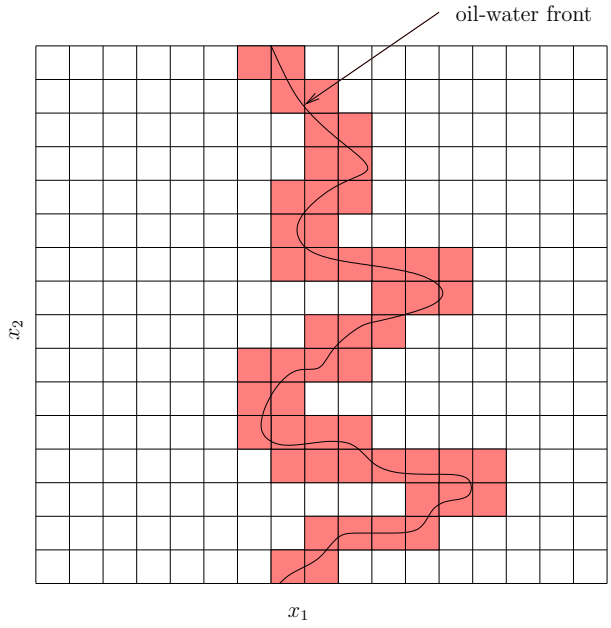


Figure 5.22: Diagram illustrating the fact that the MSFEM basis functions need only be updated frequently in a region near the oil-water front. Only in the coarse grid cells shown will the saturation be changing rapidly and behind it will be changing much more slowly.

5.5 Extension to Non-Periodic Problems

In the multiscale analysis so far we have assumed that all small scale fluctuations are periodic with respect to the fast variable $\boldsymbol{y}$. However, this is clearly a restrictive assumption that will not hold for all permeabilities. Nevertheless, the assumption is an integral part of our framework and so we restrict our attention to permeabilities with two distinct length scales for which the fluctuations can be well-approximated as being locally periodic. Given such a permeabililty, the method we describe below will give us a permeability which approximates the original and also has periodic oscillations.

For the moment we assume that our permeability $K(\boldsymbol{x})$ is a periodic function on a unit square $[0,1] \times [0,1]$, i.e. our whole domain. We expand K into its Fourier series

$$K = \sum_{\boldsymbol{r} \in Z^2} \hat{K}(\boldsymbol{r}) \exp(2\pi i \boldsymbol{r} \cdot \boldsymbol{x}) \ , \tag{5.101}$$

$i = \sqrt{-1}$, $\boldsymbol{r} = (s_1, s_2)$. Choose $0 < \epsilon = 1/E < 1$, reference wavelength, E integer. Let

$$\Lambda_E = \{\boldsymbol{r}; |s_j| \leq \frac{E}{2}, 1 \leq j \leq 2\}, \ \Lambda'_E = Z^2 - \Lambda_E \ . \tag{5.102}$$

Then,

$$K = K^{(l)} + K^{(s)} \tag{5.103}$$

where

$$K^{(l)} = \sum_{\boldsymbol{r} \in \Lambda_E} \hat{K}(\boldsymbol{r}) \exp(2\pi i \boldsymbol{r} \cdot \boldsymbol{x}) \tag{5.104}$$

$$K^{(s)} = \sum_{\boldsymbol{r} \in \Lambda'_E} \hat{K}(\boldsymbol{r}) \exp(2\pi i \boldsymbol{r} \cdot \boldsymbol{x}) \ . \tag{5.105}$$

Clearly, the component $K^{(l)}$ corresponds to the large scale permeability, and $K^{(s)}$ corresponds to the small scale permeability field. Here the superscripts s and l stand for small-scales and large-scales respectively. For each $\boldsymbol{r}$, write as

$$\boldsymbol{r} = E\boldsymbol{r}^{(s)} + \boldsymbol{r}^{(l)} \ . \tag{5.106}$$

Based on this, further decompose $K^{(s)}$ as

$$\begin{aligned} K^{(s)} &= \sum_{\boldsymbol{r}\in\Lambda'_E} \hat{K}(\boldsymbol{r})\exp(2\pi i\boldsymbol{r}\cdot\boldsymbol{x}) && (5.107)\\ &= \sum_{E\boldsymbol{r}^{(s)}+\boldsymbol{r}^{(l)}\in\Lambda'_E} \hat{K}(E\boldsymbol{r}^{(s)}+\boldsymbol{r})\exp(2\pi i(E\boldsymbol{r}^{(s)}+\boldsymbol{r})\cdot\boldsymbol{x}) && (5.108)\\ &= \sum_{\boldsymbol{r}\neq 0}\left(\sum_{\boldsymbol{r}'\in\Lambda_E} \hat{K}(E\boldsymbol{r}+\boldsymbol{r}')\exp(2\pi i\boldsymbol{r}'\cdot\boldsymbol{x})\right)\exp(2\pi i\boldsymbol{r}\cdot(E\boldsymbol{x}) && (5.109)\\ &= \sum_{\boldsymbol{r}\neq 0} \hat{K}^{(s)}(\boldsymbol{r},\boldsymbol{x})\exp(2\pi i\boldsymbol{r}\cdot\frac{\boldsymbol{x}}{\epsilon}) && (5.110)\\ &= K^{(s)}\left(\boldsymbol{x},\frac{\boldsymbol{x}}{\epsilon}\right) && (5.111) \end{aligned}$$

where the coefficient $\hat{K}^{(s)}(\boldsymbol{r},\boldsymbol{x})$ contains Fourier modes lower than $E/2$ only. Thus, we can decompose a periodic function formally into a two-scale function with periodic structure:

$$K = K^{(l)}(\boldsymbol{x}) + K^{(s)}\left(\boldsymbol{x},\frac{\boldsymbol{x}}{\epsilon}\right)\ . \tag{5.112}$$

More generally, by using a partition of unity, i.e. for a family of smooth cut-off functions $\{\phi_j\}_{j=1}^J$ such that

$$\phi_j \in C_0^1([0,1]^2) \tag{5.113}$$

$$0 \le \phi_j \le 1 \tag{5.114}$$

$$\sum_{j=1}^{J}\phi_j = 1 \tag{5.115}$$

we can decompose K as

$$K = \sum_{j=1}^{J}\phi_j K \equiv \sum_{j=1}^{J} K_j\ . \tag{5.116}$$

We can then treat K_j as a periodic function and use the same method described above to decompose the function into large and small scales. Thus, we can describe the given permeability K in the generic form

$$K = K^{(l)}(\boldsymbol{x}) + K^{(s)}(\boldsymbol{x},\boldsymbol{y}) \tag{5.117}$$

where $K^{(s)}(\boldsymbol{x},\boldsymbol{y})$ is a periodic function of period 1 in $\boldsymbol{y}$. We can use a coarse grid with size H

to resolve low frequency components of wavelength larger than ϵ and use a fine grid with size h to resolve high frequency components of wavelength smaller than ϵ. With this form for the permeability we can now apply our multiscale method for upscaling. To eliminate problems of edge effects from applying the partition of unity, we use an oversampling method. In this case, the original permeability is extended to a slightly larger domain and the method is then applied to that domain.

We now demonstrate the ability of the above method to capture the fluctuations correctly. We take an example permeability and apply the above method to it. Since the permeability is a strictly positive quantity, i.e. $K > 0$ throughout the domain, we find it better to apply the method to the log of the permeability, i.e. to $u = \log(K)$. By applying it to this and then taking the exponential, we are guaranteed that the resulting approximation is also strictly positive.

To demonstrate the method we show some results for two different types of permeability. The first is a non-layered example. The original log permeability is shown in the first plot of figure 5.23. This permeability was generated using the GSLIB package on a 256×256 uniform grid. This is then interpolated (bilinear interpolation) to give a 1024×1024 permeability. The interpolation is done because the variation in the original permeability is very strong and even on a 256×256 grid can appear discontinous as one crosses the layers of high/low permeability. With this new permeability the above method is then applied and the resulting reconstruction is shown in the second plot of figure 5.23. For this, the cut-off wavelength was $\epsilon = 1/32$. Each periodic wave was reconstructed using a 16×16 subgrid. As can be seen from the plots, the reconstruction is very close to the original permeability. Statistics for the original and reconstruction are shown in table 5.7. Since it is difficult to distinguish the solutions at this magnification, we show a close up of the reconstruction in figure 5.24. From this, one can see the boundaries at which we force the solution to be locally periodic.

Similarly, we show results for a layered permeability in figure 5.25. The results show again that the reconstruction works well.

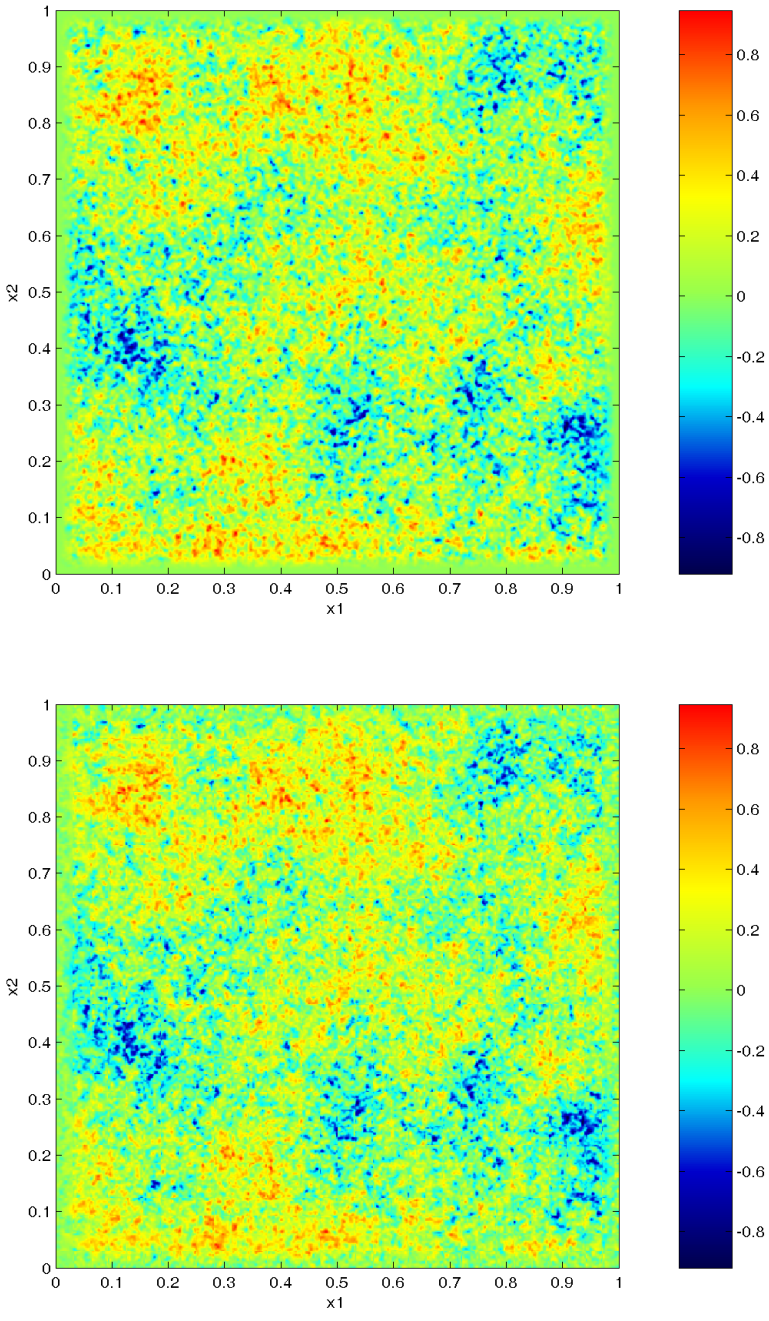


Figure 5.23: Original log permeability and "reparametrized" log permeability for a non-layered case.

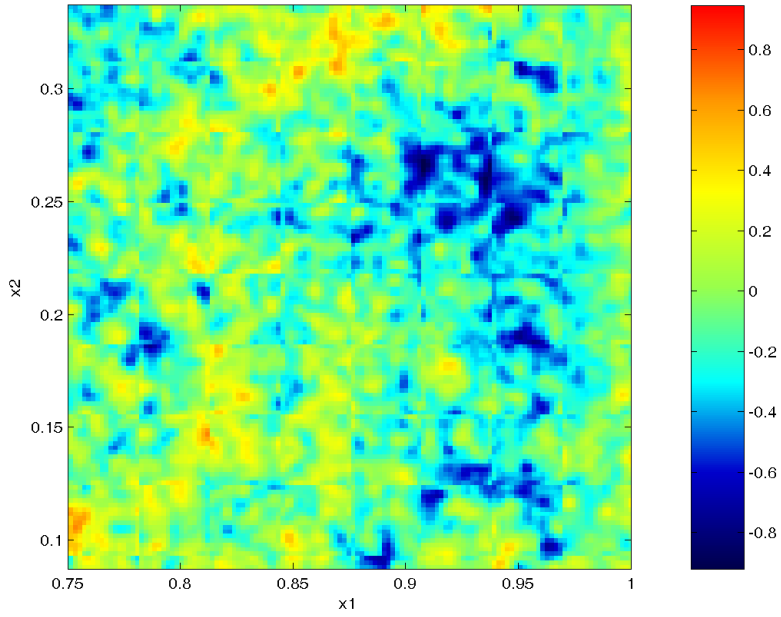


Figure 5.24: Close up of the reconstructed log permeability.

	Original	Reparametrized
max	0.947	0.968
min	-0.920	-1.005
mean	0.053	0.053
variance	0.053	0.052
l_2 error	–	0.152
max error	–	0.96

Table 5.7: Statistics for the original and reconstructed log permeability in the non-layered case.

	Original	Reparametrized
max	1.074	1.022
min	-0.914	-0.943
mean	0.0693	0.0693
variance	0.0642	0.0639
l_2 error	–	0.05
max error	–	0.321

Table 5.8: Statistics for the original and reconstructed log permeability in the layered case.

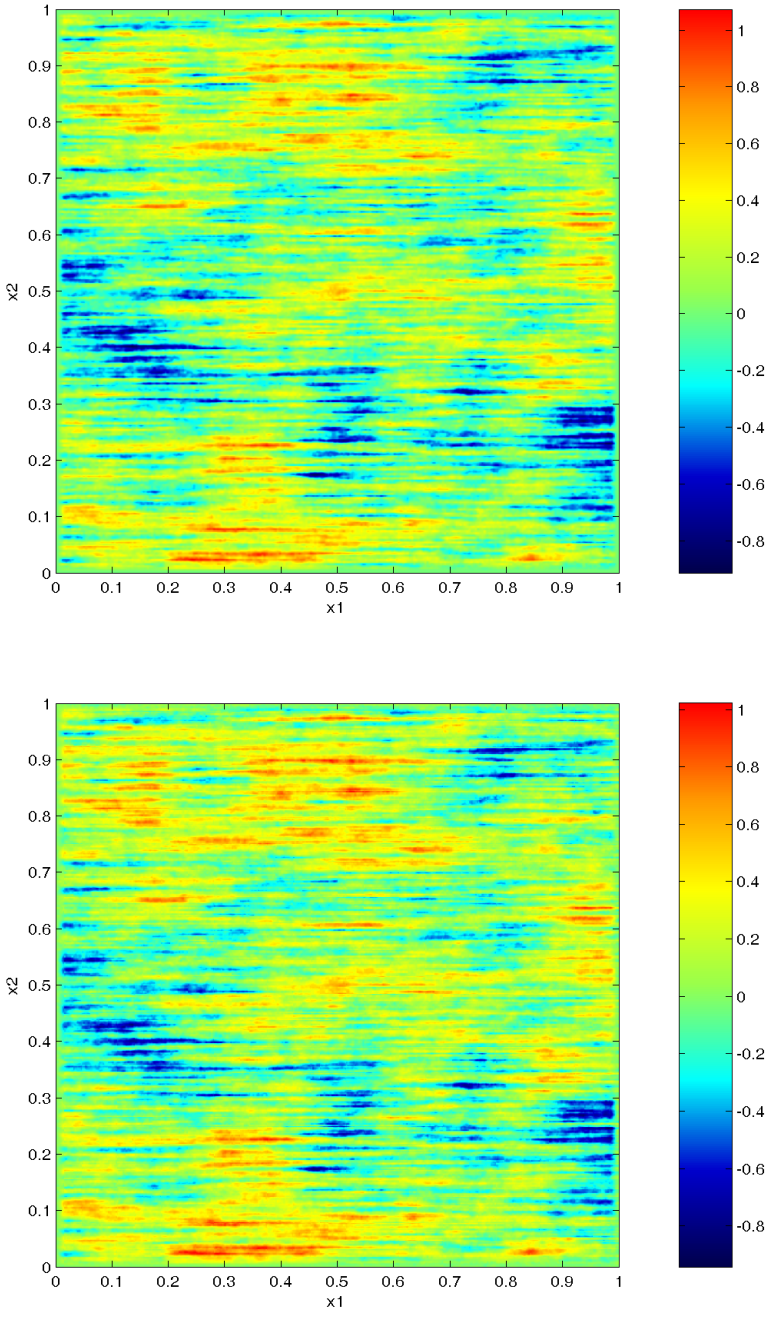


Figure 5.25: Original log permeability and "reparametrized" log permeability for a layered case.

Chapter 6

Numerical Results

6.1 Overview

In this chapter we present numerical results for the scheme proposed in the previous chapter. We first demonstrate that the scheme is first order accurate in capturing the average saturation for the single-phase case for periodic permeabilities. We perform simulations in much the same manner as that done in section 5.3.5 of Chapter 5 for the core-plug model boundary conditions. After this we then demonstrate the efficacy of our upscaling scheme for non-periodic permeabilities, using the method proposed to approximate the non-periodic oscillations into locally periodic ones. In this case, to evaluate the performance the upscaling scheme we compare the fractional flow curves, as described in section 3.3 from the resolved scheme and the upscaling scheme. We find that our upscaling scheme very accurately computes this important quantity. In addition, we also perform timing tests to see how well the scheme compares in run-time to the resolved calculation and give an analysis of the timings, with a view that effective implementation may increase the performance benefits of the upscaling scheme.

The chapter is laid out following outline described above.

6.2 Periodic Permeability Field

To demonstrate our method converges, we present some examples where the permeability is a prescribed two-scale function. We do this for both single-phase using an analysis similar to that given in section 5.3.5. We do not use the two-phase results here since the shock-formation in that case reduces our ability to analyse the formal convergence rate of the

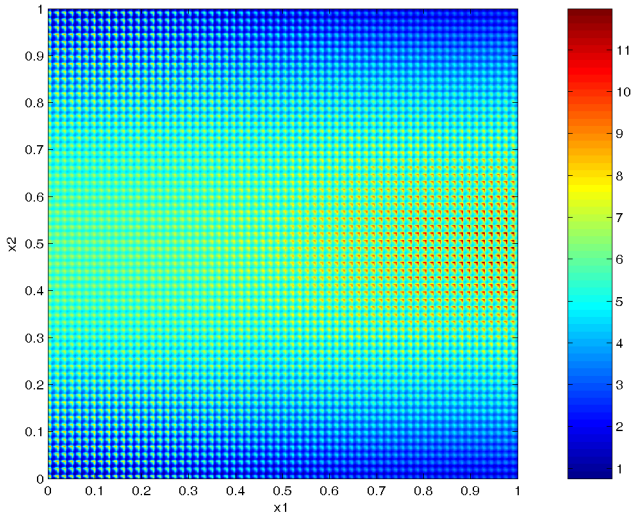


Figure 6.1: Permeability used in the periodic case.

scheme. We use only the core-plug model boundary conditions in this case. The sections that follow describe in detail the results obtained.

6.2.1 Single-Phase Results

To test the scheme in this case, we prescribe a permeability field with fast periodic oscillations. We use

$$K(\boldsymbol{x},\boldsymbol{y}) = 15x_2(1.0 - x_2) + \frac{2 + P(x_1,x_2)\sin(2\pi y_1)}{2 + P(x_1,x_2)\cos(2\pi y_2)} + \frac{2 + \sin(2\pi y_2)}{2 + P(x_1,x_2)\cos(2\pi y_1)} \tag{6.1}$$

where

$$P(\boldsymbol{x}) = 1 + \frac{1}{2}\cos(\pi x_1)\cos(2\pi x_2) \tag{6.2}$$

We set the small scale parameter $\epsilon = 1/64$. Figure 6.1 shows the permeability field. As can be seen, the permeability has rapid oscillations in the horizontal and vertical directions, with the magnitude of the oscillations greatest in the center. A similar model for the permeability was used in [19] in testing the convergence of MSFEM. We set up the boundary conditions for the pressure equation exactly as described in section 3.3.1. Since the pressure is uncoupled from the saturation in the single-phase case, we need only solve for this once at the start of the simulation. The first test we perform is to check the convergence rate as the number of

N	l^2 error	max error
16	0.0780	0.3725
32	0.0556	0.2793
64	0.0460	0.2239

Table 6.1: Errors in the multiscale reconstruction in the single-phase case using different grids.

N	l^2 error	max error
16	0.0745	0.2204
32	0.0419	0.1155
64	0.0215	0.0595

Table 6.2: Errors in the homogenized in the single-phase case using different grids.

N	l^2 error	max error
16	0.2098	0.4014
32	0.1216	0.2416
64	0.0655	0.1636

Table 6.3: Errors in the velocity field in the single-phase case using different grids.

coarse grid points is increased. We do this in the same manner as that used in section 5.3.5 when we tested the scheme for the saturation seperately, i.e. by comparing the results with resolved calculations for different coarse grid. Again, we use 16×16, 32×32 and 64×64 coarse grids. We keep the sub-grids the same in each of these cases, using 32×32 sub-grid points. The initial data for the saturation is given by (5.66). We evolve the saturation up until time $t = 0.1$ and then compute the l^2 and infinity norms of the error in the multiscale reconstruction and the average. Figure 6.2 shows the resolved saturation and the average computed from this. Figure 6.3 shows the multiscale reconstruction for the different coarse grids. We see that the scheme captures the multiscale features well. However, perhaps due to compounding of errors, the convergence rate for these grids is not yet first order. The slope of the log-log error line, shown in Figure 6.5 is only about 0.38. Since both parts of the scheme (saturation and velocity solver) are formally first order, we suspect that the scheme would eventually show this as the number of coarse grid points is increased. However, due to the limitations of our computer resources, we were unable to verify this. The average, however, is clearly first-order accurate, as can be seen from the slope of the line on the log-log error plot.

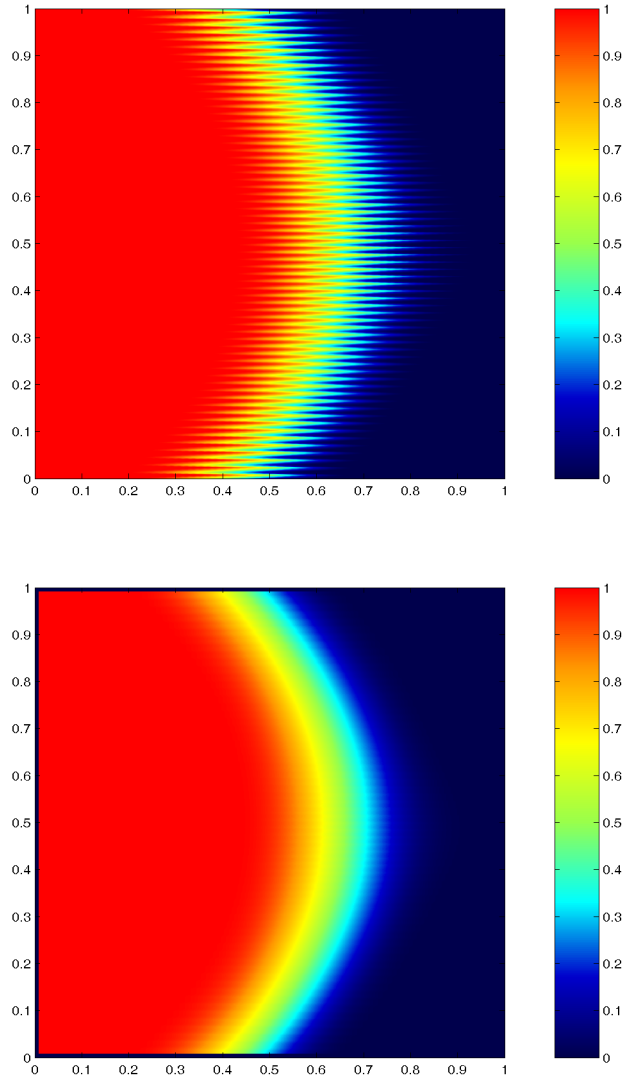


Figure 6.2: Density plot for the "exact" solution for the saturation in the single-phase case, resolved and average.

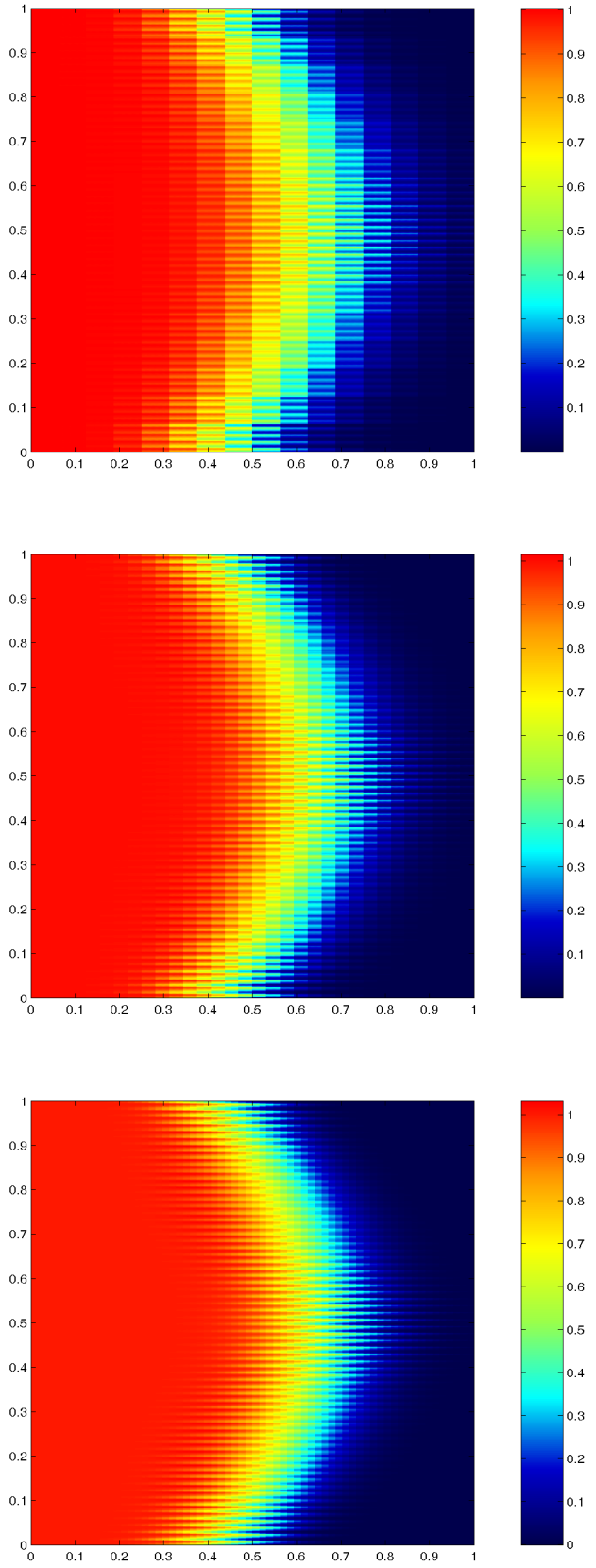


Figure 6.3: Upscaled transport combined with MSFEM on different grids for the single-phase case, reconstructed multiscale solution.

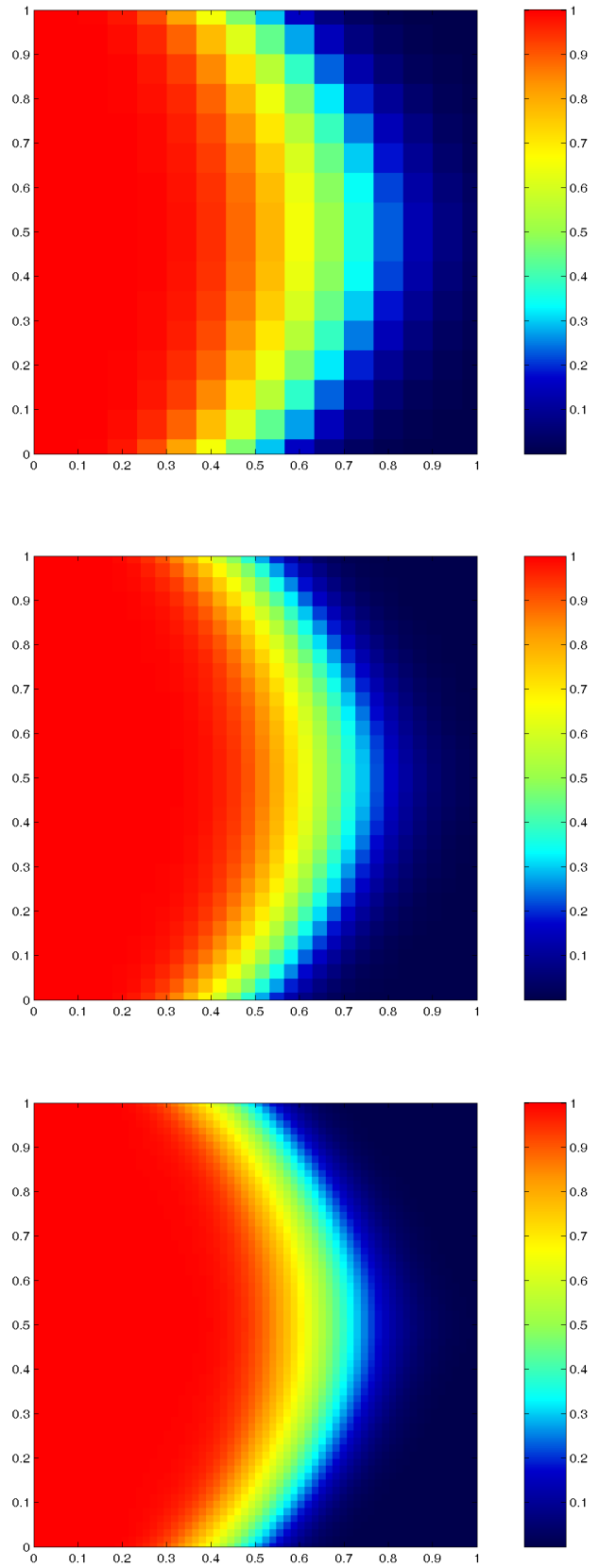
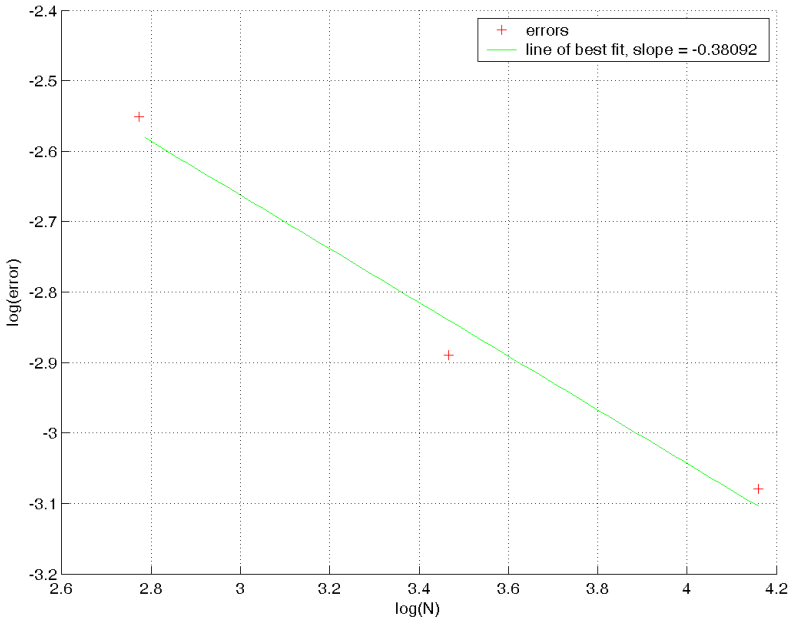


Figure 6.4: Upscaled transport combined with MSFEM on different grids for the single-phase case, average solution.



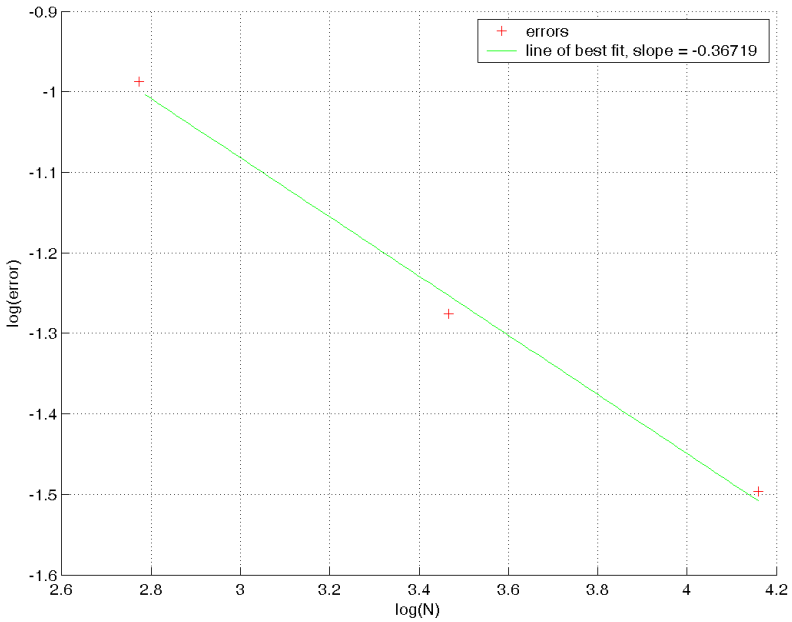


Figure 6.5: Log-log error plots for the l^2 and max norm error respectively in the reconstructed multiscale solution for the single-phase case.

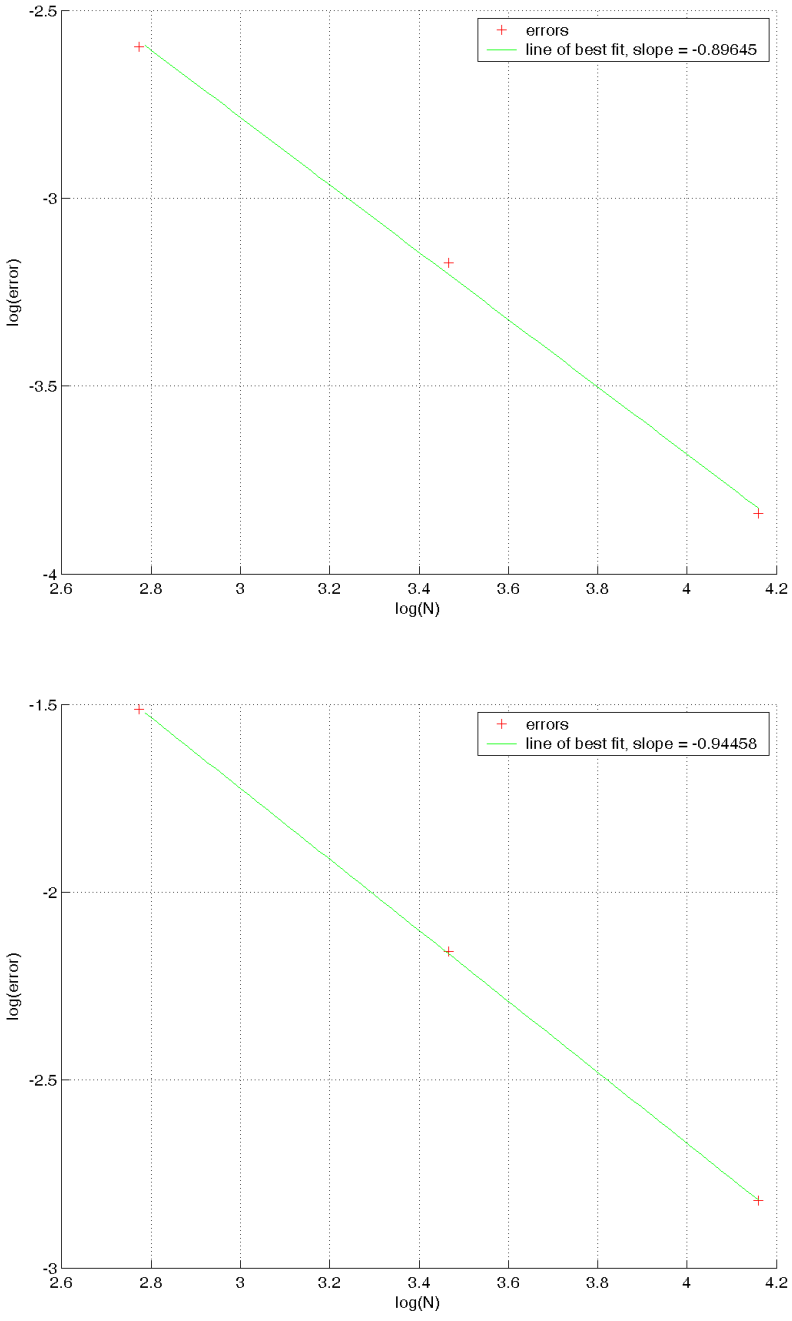


Figure 6.6: Log-log error plots for the l^2 and max norm error respectively in the homogenized solution for the single-phase case, demonstrating first-order convergence.

6.3 Non-Periodic Examples

We now apply all the preceeding numerical methods for the upscaling problem to a case where the permeability field is a given non-periodic quantity. We apply the method to the permeability fields shown in Chapter 5 in figures 5.23 for 1-phase and 2-phase computations. As before, we compare the resulting solutions for the average saturation with those computed using resolved calculations. In this case, we cannot hope to obtain the same kind of convergence rates that we obtained in the previous sections where all the fluctuations had a definite periodic structure. Therefore, as a measure of accuracy, we compute the fractional flow curves that were described in Chapter 2, given by equation (3.39). As mentioned in that section, this is a feature of interest to engineers when evaluating a reservoir simulation and any upscaling scheme should aim to reproduce this accurately.

For the tests we use the same boundary conditions and initial data as those used in section 5.3.5.

6.3.1 Single-Phase Results

Figure 6.7 shows the logarithm of permeability field. This is given on a 256×256 grid, as

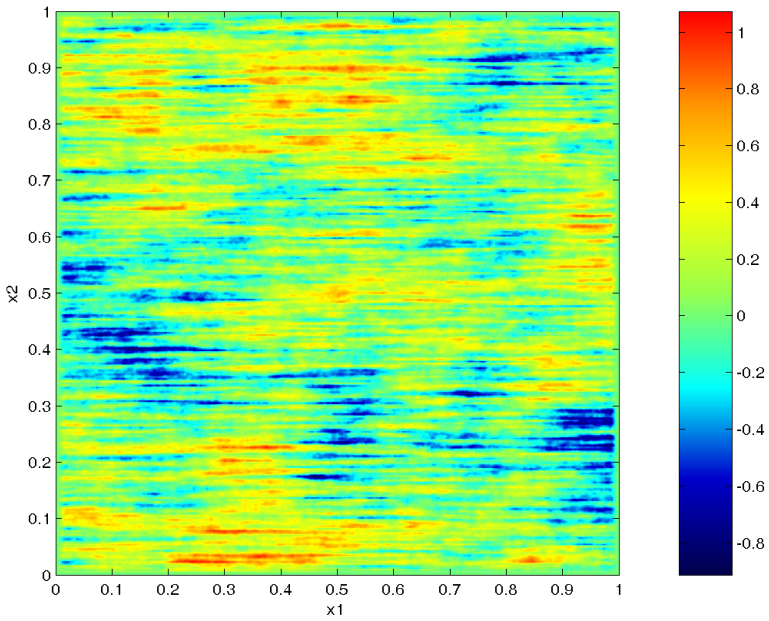


Figure 6.7: Layered permeability used.

described in section 5.5. For the resolved computations, we use a 1024×1024 grid which was found to be sufficient to fully resolve flow features. For the upscaled computations, including the permeability re-parameterization, we use a 64×64 coarse grid with 16×16 sub-grid points. This configuration gives us a "scale-up" factor of 8 in each direction (since the average solution is given on a 64×64 grid). Figure 5.23 shows the re-parametrized permeability field obtained by applying the method described in section 5.5 to this permeability. This is almost indistinguishable from the original permeability.

We first compare the velocity fields computed from the resolved scheme and the upscaling (MSFEM) method. Figure 6.8 shows the horizontal (x_1) component of the velocity field as computed by the different methods. The details of the velocity are captured well, with layers computed accurately. Because of the coarser grid in the MSFEM computations, there are some slight edge effects which can be seen. Note, however, that this velocity field shown is not used directly in the upscaled computations, but rather it's average and the locally periodic velocity are used. The average component of the horizontal velocity as computed from the resolved scheme and MSFEM are shown in figure 6.9. The agreement between these is clearly very good.

Figures 6.10 and 6.11 show the corresponding results for the vertical component of the velocity field. In this case one can seen again that the velocity field is captured well, though not as well as for the horizontal component, especially for the resolved features. However, since the main flow is in the horizontal direction this has only a minor effect on the saturation computations.

The above velocity fields are now used to advance the saturation. Since the aim of the method is to accurately compute the average, we compare the average saturations computed from the resolved calculations and the upscaling scheme at several times. The initial saturations for both methods is shown in figure 5.9. Figure 6.12 shows the resolved calculations at time $t = 0.17$. Note the amount of "fingering" of the saturation front, which is due to the amount of layering in the permeability and velocity fields. Figure 6.13 shows the average saturation at this time computed from the resolved computations and upscaled scheme. From these one can see that the upscaling scheme is accurately capturing the average. In addition to these we also show the results that are obtained if one ignores the

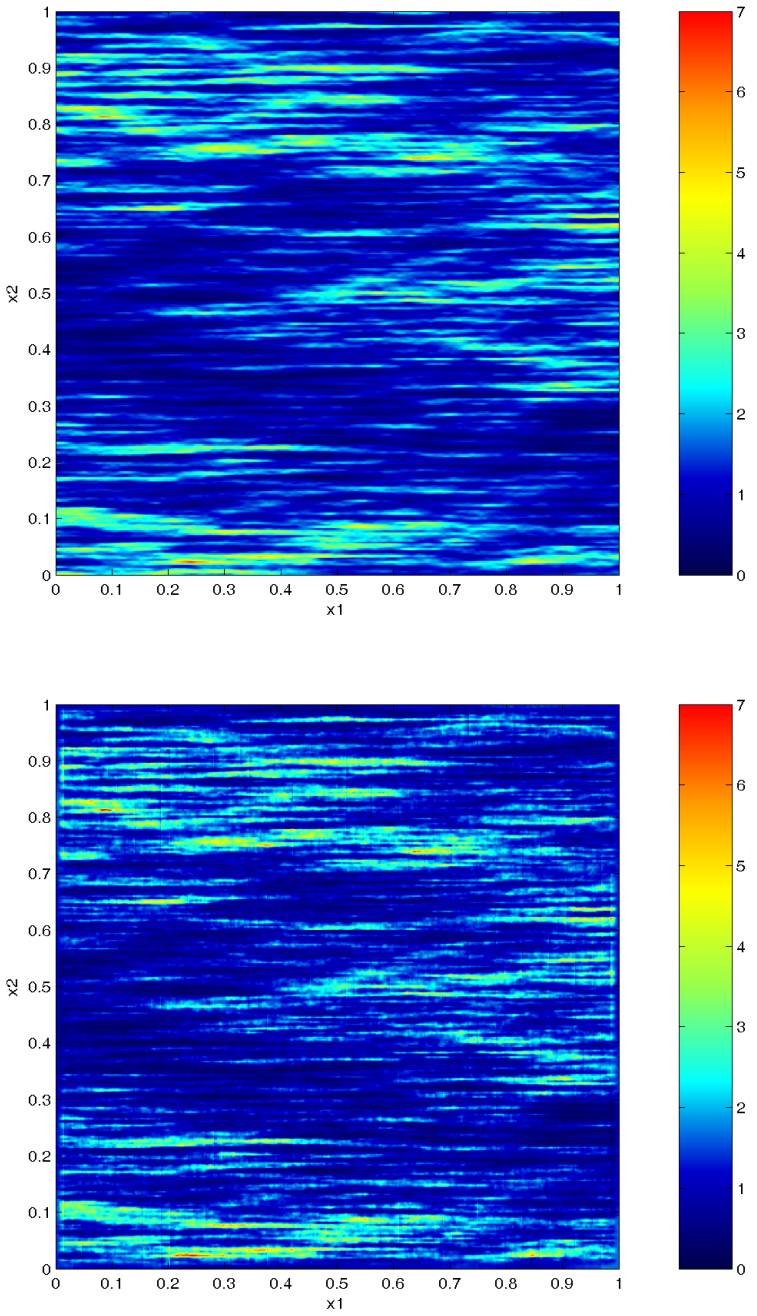


Figure 6.8: Comparison of the horizontal components of the velocity field computed using resolved scheme (top), and MSFEM (bottom).

interaction terms when computing the average, i.e. we solve

$$\frac{\partial \overline{S}}{\partial t} + \overline{\boldsymbol{v}} \cdot \nabla_x f(\overline{S}) = 0 \tag{6.3}$$

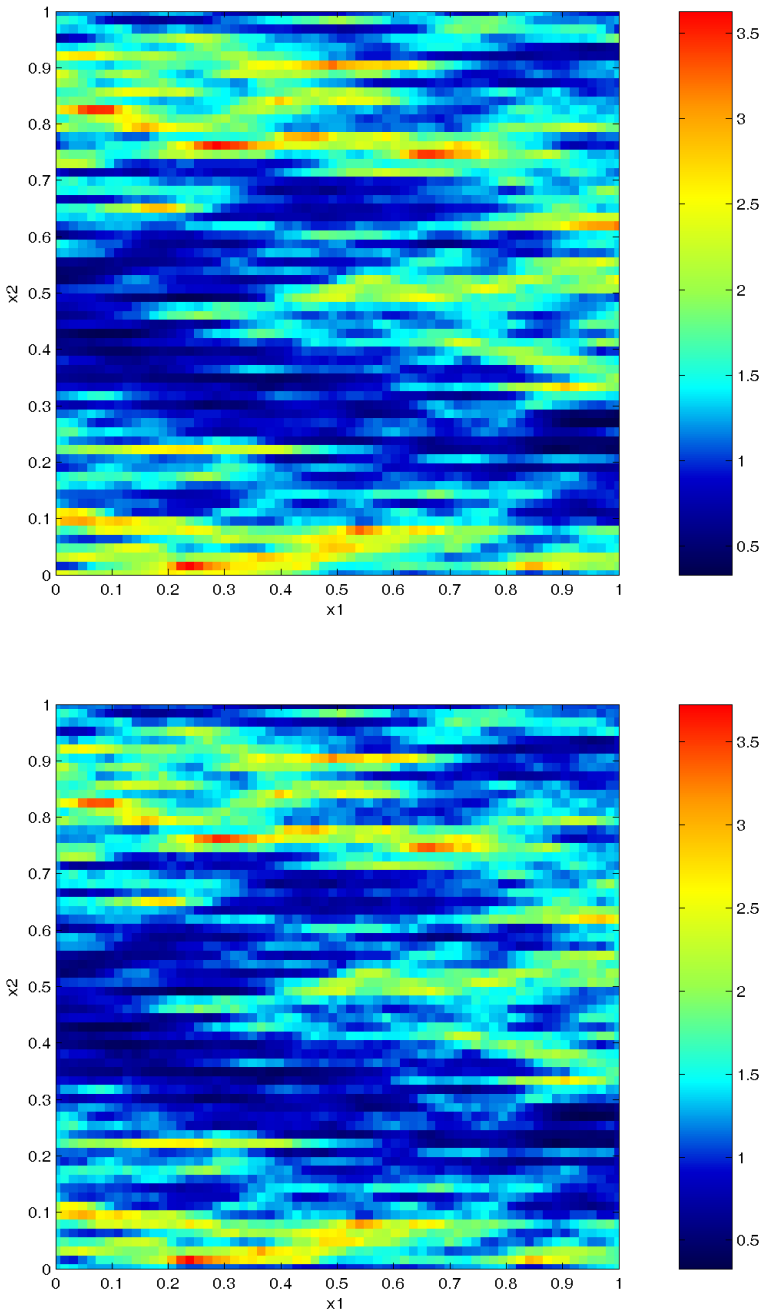


Figure 6.9: Comparison of the average horizontal components of the velocity field computed using resolved scheme (top), and MSFEM (bottom).

for the average saturation $\overline{S}$ where $\overline{\boldsymbol{v}}$ is computed using the MSFEM. One can see that in this case, and more clearly at subsequent times, that the average saturation is not being moved correctly in this case. This confirms that including the interaction terms is vital in

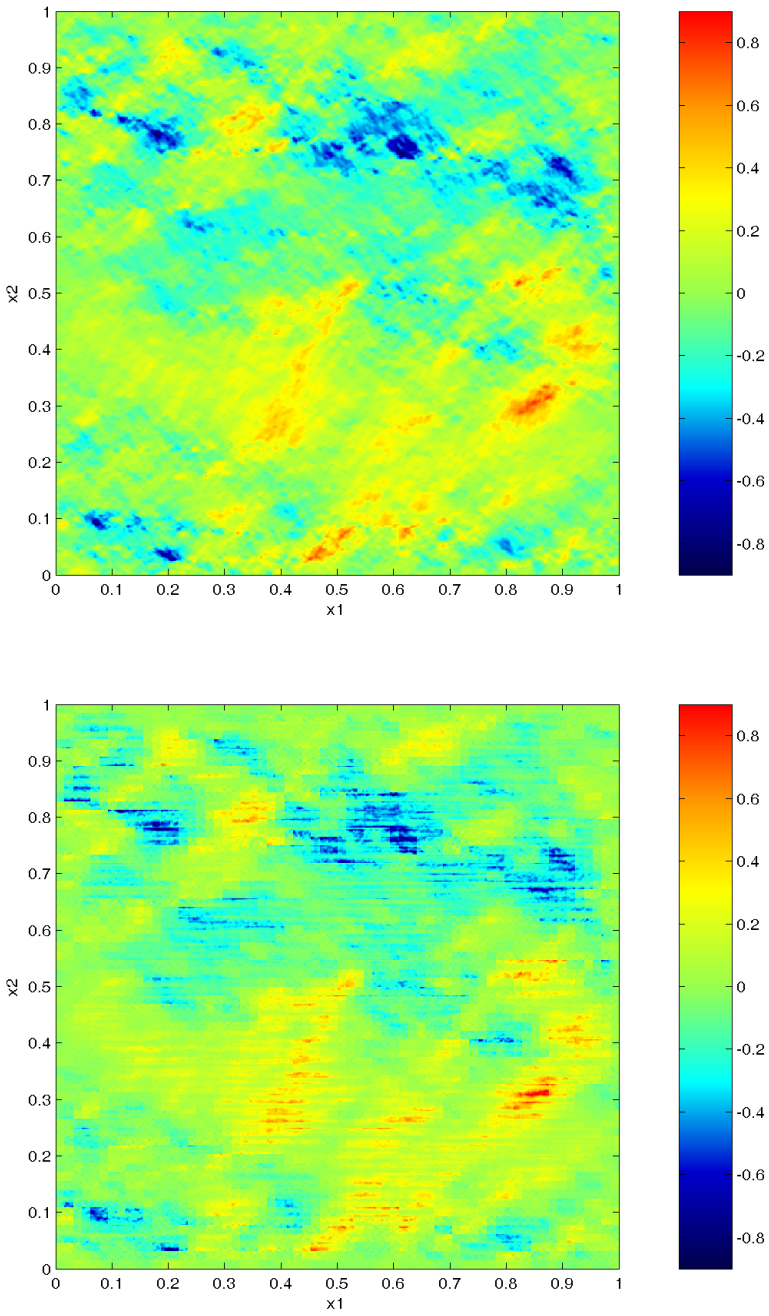


Figure 6.10: Comparison of the vertical components of the velocity field computed using resolved scheme (top), and MSFEM (bottom).

computing the average correctly. It seems for this example, that if the interaction terms are ignored, that the bulk of the saturation is moved too slowly.

Figure 6.14 shows the average saturation at time $t = 0.3$ computed from the resolved

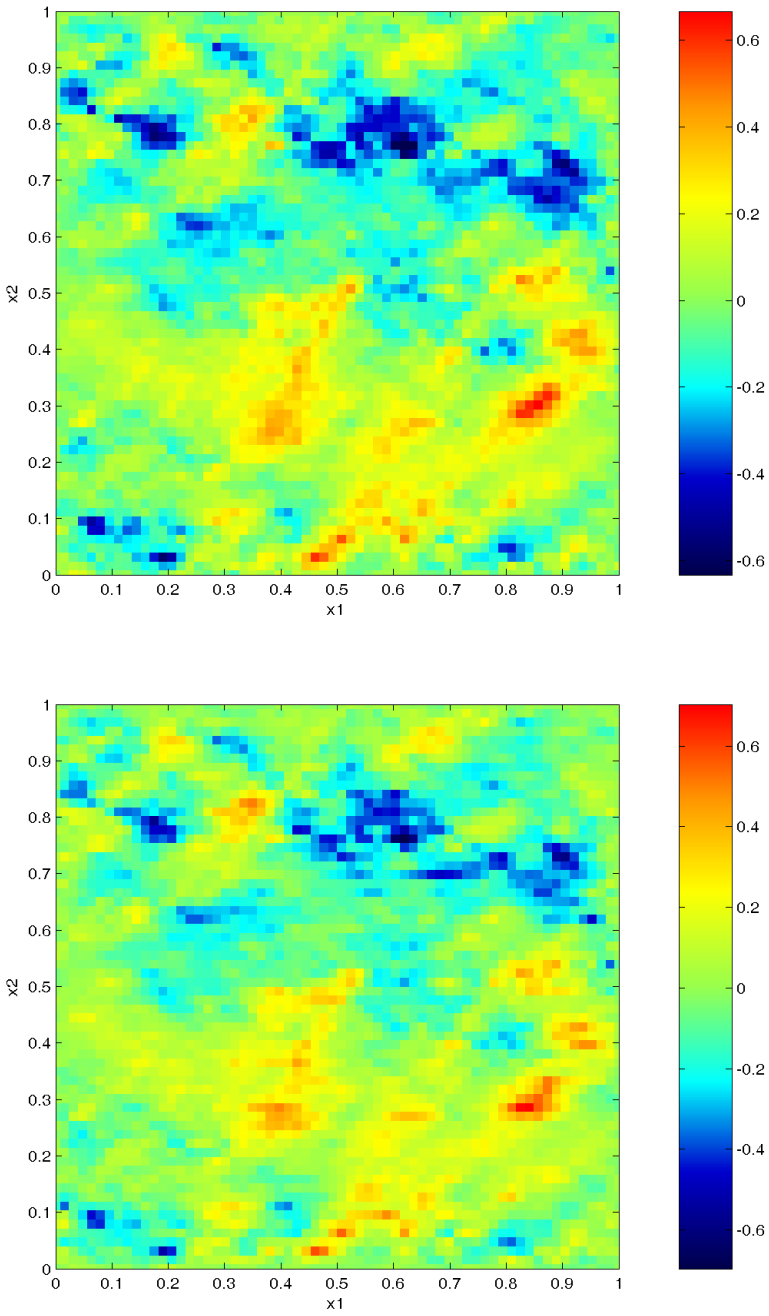


Figure 6.11: Comparison of the vertical horizontal components of the velocity field computed using resolved scheme (top), and MSFEM (bottom).

computations and upscaled scheme.

Figure 6.15 shows the average saturation at time $t = 0.45$ computed from the resolved computations and upscaled scheme.

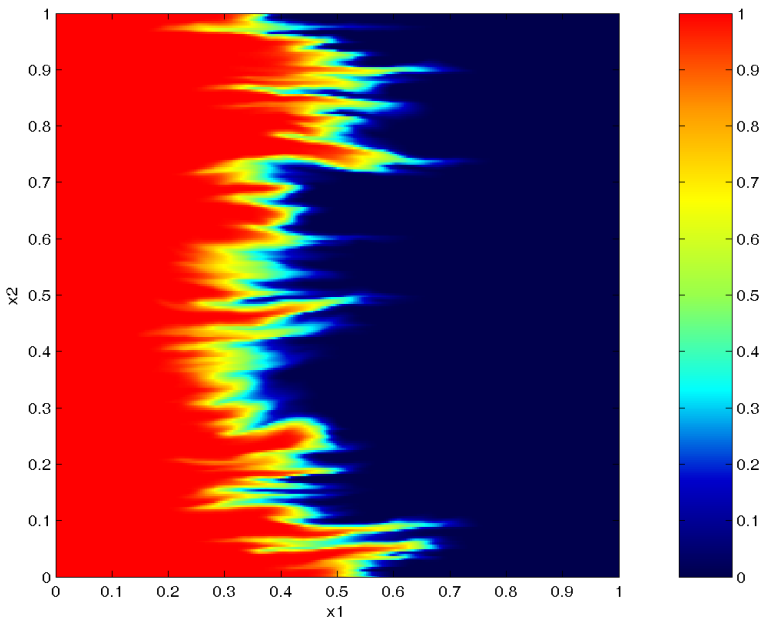


Figure 6.12: The resolved saturation at $t = 0.17$.

Figure 6.16 shows the fractional flow curves computed using the resolved computations, the upscaled method and the naive approach described above. As with the saturation plots one can see that the upscaled computations accurately capture the true fractional flow. Again, if one ignores the interaction terms in the upscaling method then the results are much poorer. One can see that for each of the methods, the time at which water reaches $x = 1$ (the breakthrough time) is similar for each. However, for the naive method, since the bulk of the saturation is moved too slowly, the fractional flow curve is too high after the breakthrough time and remains far from the true fractional flow for all subsequent times. For the upscaled scheme, the fractional flow curve follows the true one closely for most of the time. There are some slight differences at later times, which are most likely due to the fact, mentioned above, that the average velocity field computed from MSFEM is not exactly divergence-free.

6.3.2 Two-Phase Results

For the two-phase flows we perform exactly the same analysis as was done above for the one-phase case. In this case, since it has already been demonstrated that MSFEM accurately

captures the velocity field, we skip the comparisons of the velocity fields. Figure 6.17 shows the resolved computations at $t = 0.17$. For this case, there is not as much "fingering" of the saturation into the layers of high permeability as in the single-phase case but there is still quite alot of small scale features. In figures 6.18, 6.19 and 6.20 we again show the average saturations computed from the resolved, upscaled and naive methods at the times $t = 0.17,\ 0.3,\ 0.45$. From these one can see again that the upscaled method captures the average saturation accurately whilst the naive method does not move the average correctly. This is also evident from the fractional flow curves shown in figure 6.21.

6.3.3 Timing Results

A stated goal of the upscaling scheme is that computations involved should take less time to run compared to the resolved computations. We therefore performed a timing comparison for the two-phase computations shown above (note that we did not do it for the single-phase computations because in that case the velocity field does not need to be updated at each stage, which is a major burden in the computations). The timing experiments provided give only an approximate guide to the performance efficiency since they were done using a limited set of runs. In addition, it is possible that more sophisticated coding techniques could allow for further benefits. The code itself used a mixture of FORTRAN, C and C++. The experiments were performed on a Windows XP machine (using the UNIX emulator Cygwin) with a 2.4 GHz processor and 512 MB of memory. For the $t = 0.45$ results, shown above, we compare the total run times of the two compuations, and also their breakdown into the most significant contributions to these run times.

For the resolved computations, the total run time was close to 24200 seconds (about 6.75 hours). For the upscaled calculations, the total run time was 14700 seconds (just over 4 hours). Whilst this is not an order of magnitude greater it is worth pointing out the breakdown in the timing for both.

In the resolved calculations, the vast majority (more than 99%) of the time is spent in either the elliptic solver (for the velocity field updates) or else in the hyperbolic solver (saturation field updates). Between these two, the elliptic solver is the much more expensive stage, even if we use the pressure from the previous time step as the initial guess for solution at the current time step. As shown in the breakdown chart in the upper pie chart of figure 6.22, 91% of the time is in this elliptic solver stage (which includes the construction of

the linear system of equations). The remaining time is spent in the hyperbolic solver. As mentioned, the elliptic equation is solved using preconditioned conjugate gradient (PCG) method and without resorting to an even faster scheme (such as multigrid) this stage cannot be improved much. (We use a PCG method rather than a multigrid method since most multigrid packages require that the grid size be of the form $2^r + 1 \times 2^r + 1$ where r is a positive integer. Our grid is 1024×1024 since we use cell-centered pressures. PCG packages generally have no such restrictions.)

For the upscaled computations the breakdown, shown in the lower pie chart in figure 6.22, shows the elliptic parts (which includes the MSFEM and all the associated steps in computing the basis functions) takes up 21% of the run-time, the hyperbolic part (including computation of source terms but not including the streamline projection) takes up 45% of the run-time and the streamline projection computations take up 34 %. As mentioned in the section on the streamline projection, finding an efficient method for computing this was very difficult. Potentially, if one were able to reduce the run-time in this section, the upscaling scheme would be significantly improved and the overall run time for the upscaling computations could be half that of the resolved computations. In addition, in longer simulations, perhaps where one is interested in following the reservoir performance up until 90% of the oil is removed, the upscaling scheme should see further benefits due to the fact that very few basis functions will need updated and hence the elliptic step will be extremely fast. Further, in practical simulations the reservoir is likely to have significantly more than 1024×1024 grid blocks and in these cases we can expect that an upscaling scheme will have an even better scale up factor than the 8 which we used here.

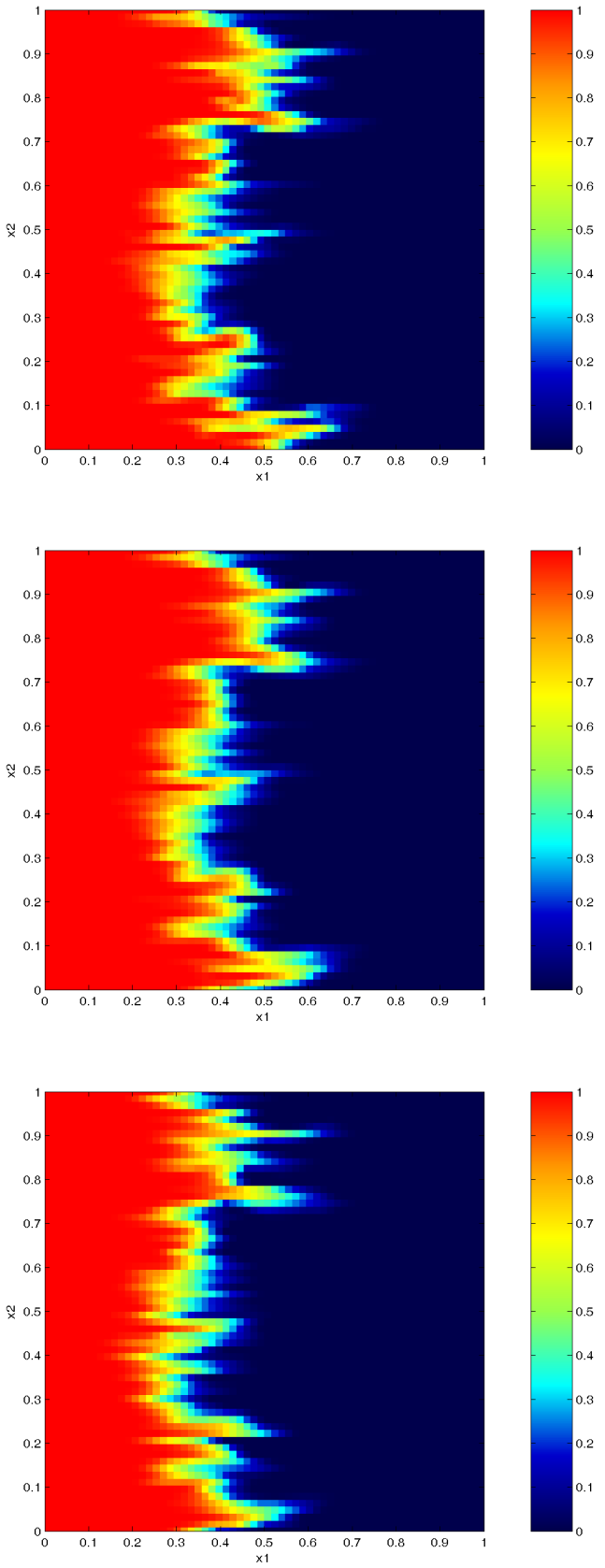


Figure 6.13: Comparison of the average saturation profiles for the single-phase case at $t = 0.17$ for the exact calculation (top), upscaled (center), naive (bottom).

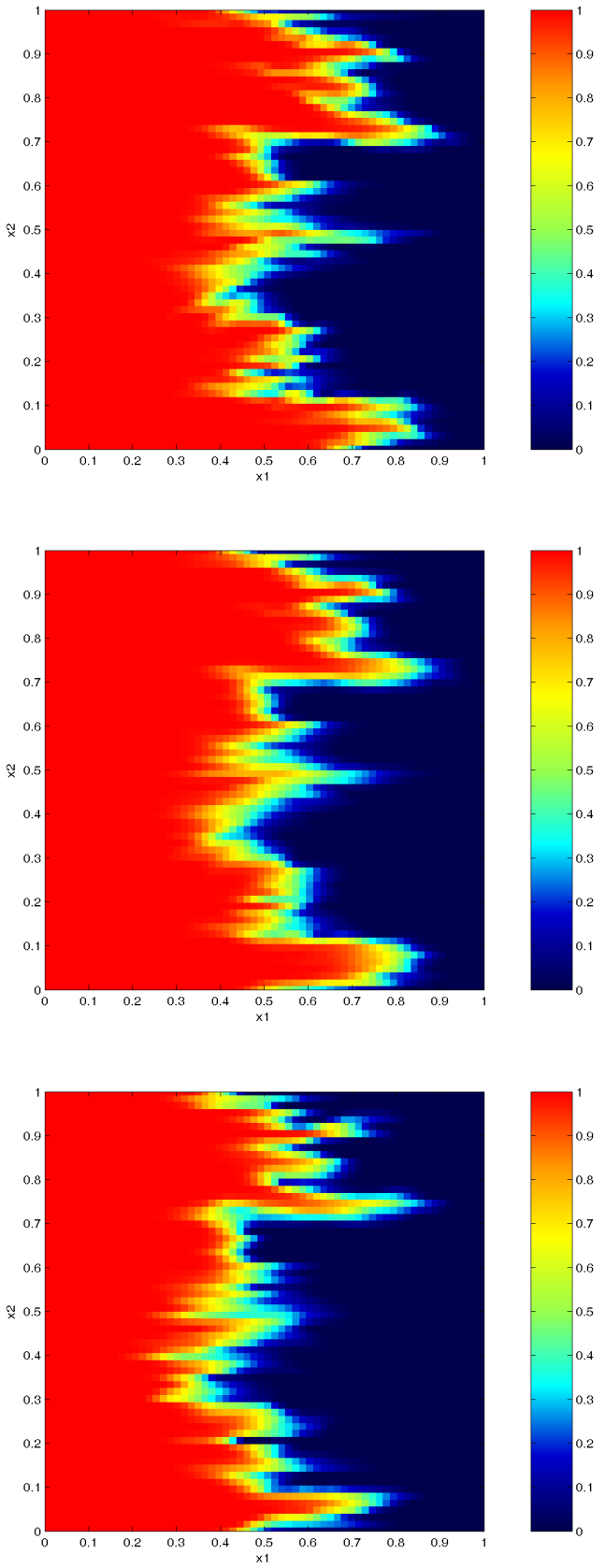


Figure 6.14: Comparison of the average saturation profiles for the single-phase case at $t = 0.3$ for the exact calculation (top), upscaled (center), naive (bottom).

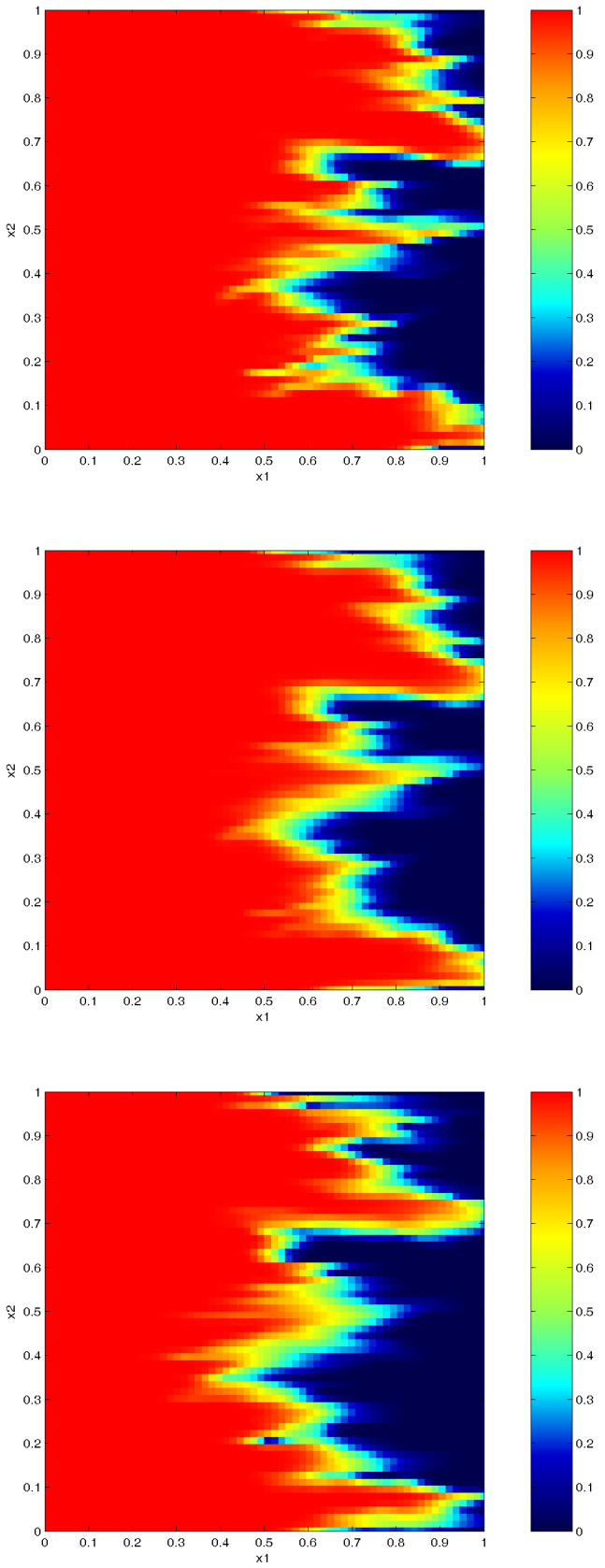


Figure 6.15: Comparison of the average saturation profiles for the single-phase case at $t = 0.45$ for the exact calculation (top), upscaled (center), naive (bottom).

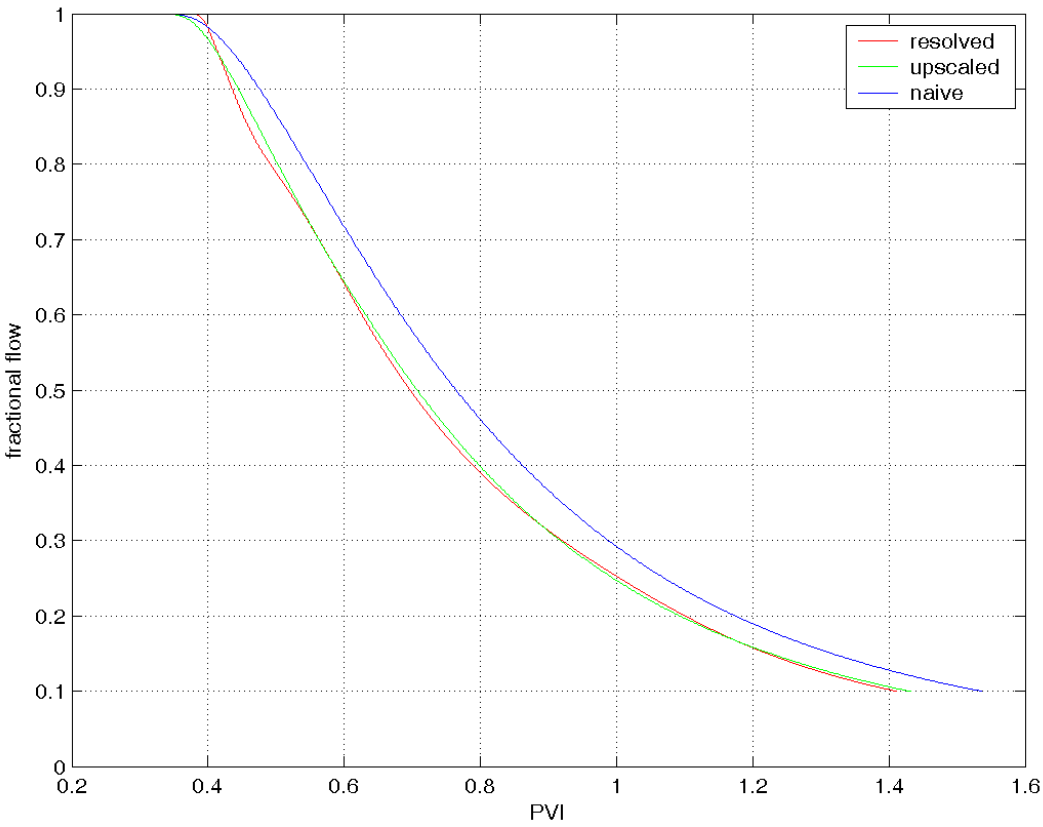


Figure 6.16: Comparison of the fractional flow curves for the single-phase case.

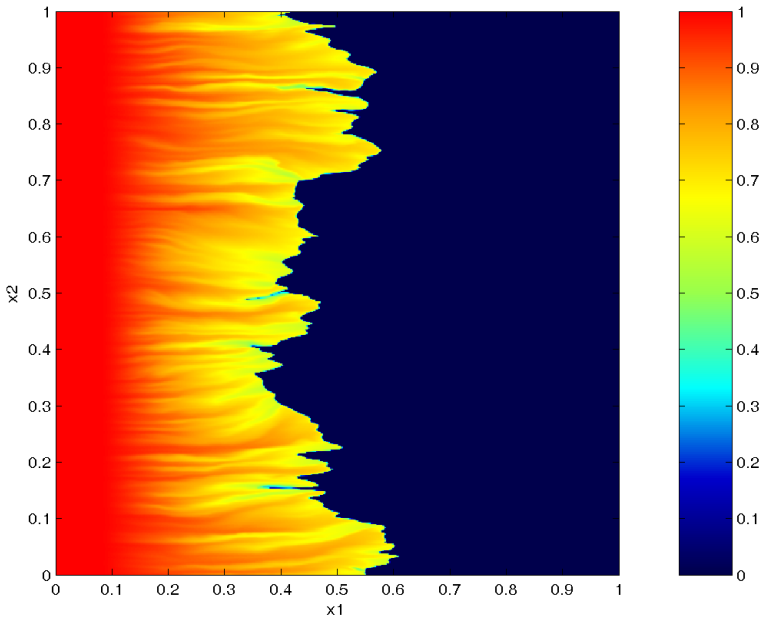


Figure 6.17: The resolved saturation at $t = 0.17$ for the two-phase case.

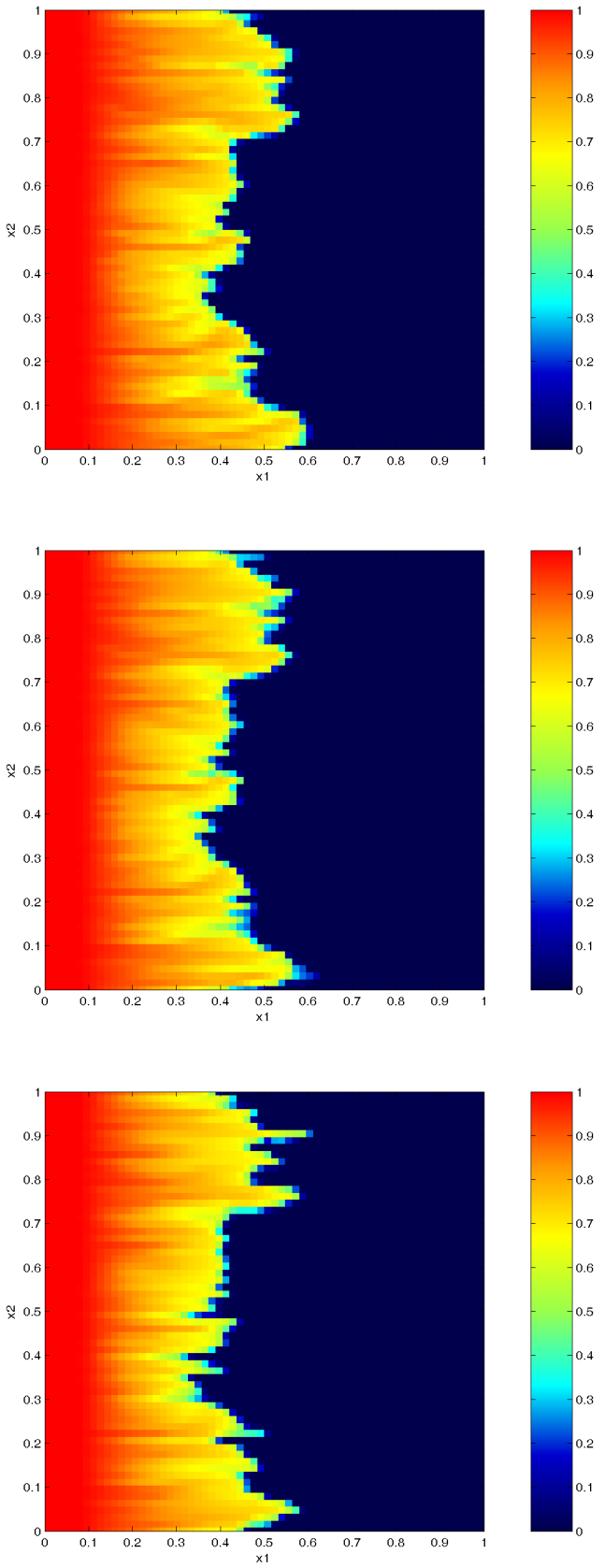


Figure 6.18: Comparison of the average saturation profiles for the two-phase case at $t = 0.17$ for the exact calculation (top), upscaled (center), naive (bottom).

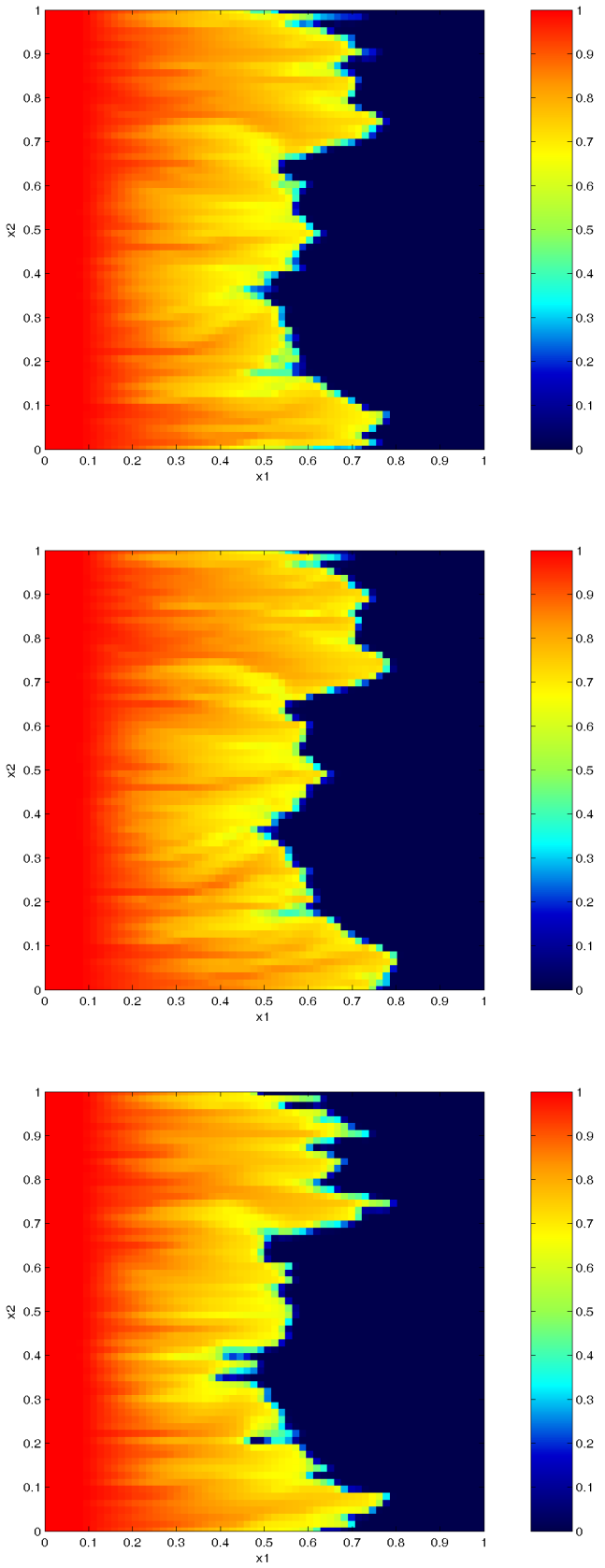


Figure 6.19: Comparison of the average saturation profiles for the two-phase case at $t = 0.3$ for the exact calculation (top), upscaled (center), naive (bottom).

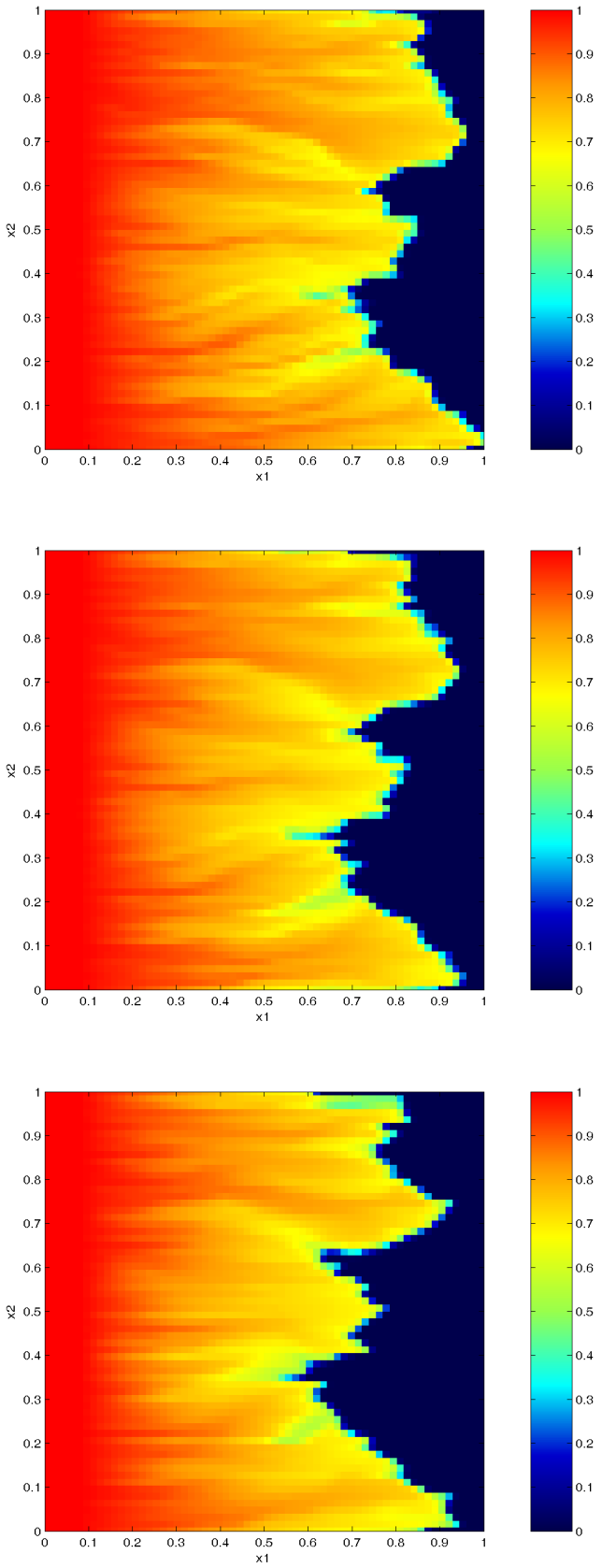


Figure 6.20: Comparison of the average saturation profiles for the two-phase case at $t = 0.45$ for the exact calculation (top), upscaled (center), naive (bottom).

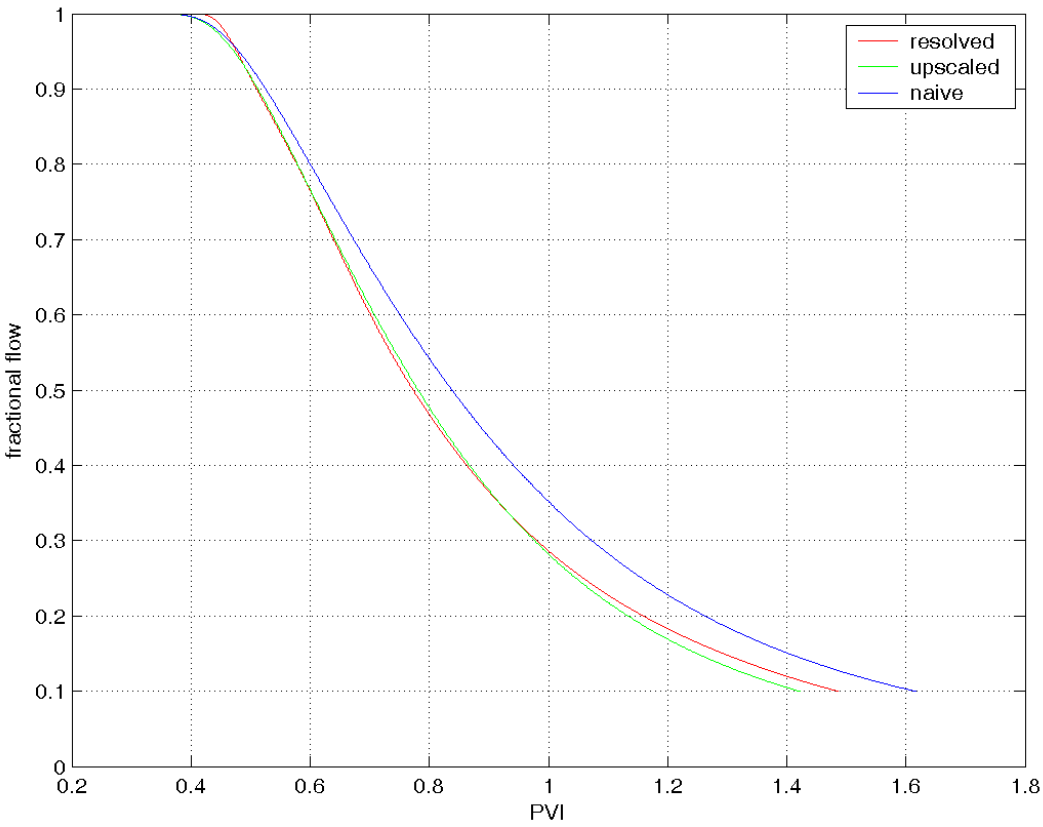


Figure 6.21: Comparison of the fractional flow curves for the two-phase case.

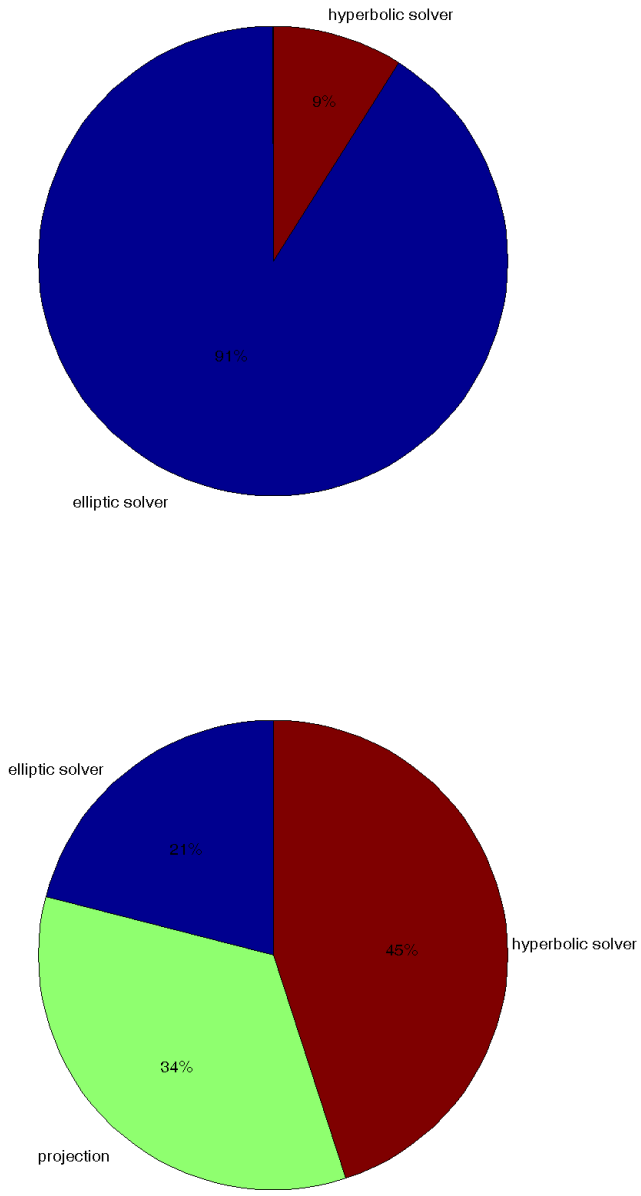


Figure 6.22: Timing breakdowns for the resolved computations (top) and upscaled computations (lower).

Chapter 7

Summary and Conclusions

In this these we have examined the need for upscaling in porous media flow simulations, in particular in modeling the saturation in an enhanced oil recovery scheme. We have presented both a framework for developing a multiscale analysis of the problem and a numerical scheme for solving the resulting upscaled equations. The multiscale analysis was facilitated by making the assumptions that the rapid spatial fluctuations in the underlying permeability were of a distinct length scale from the larger features and also that they were periodic. By manipulating the hyperbolic equation for the saturation and the introduction of a special streamline averaging which eliminates fast time scales from the problem we were able to split this equation into a pair of coupled hyperbolic equations for the average saturation and the fluctuations about this average. We then developed a novel numerical scheme by solving for only a subset of the fluctuations and then using these to approximate the large scale small scale interaction terms in the average equation. Both the average equation and the fluctuation equations were solved using standard hyperbolic solvers. This numerical scheme was demonstrated to have first order accuracy in capturing the average saturation for both one and two-phase flows in the case where a periodic velocity field was prescribed. To extend the method to the more practical case when the permeability is not periodic we used a re-parametrization of the permeability to give an approximation that was locally periodic. This was then used in the numerical scheme. In addition, we used a special variant of MSFEM to numerically evaluate the velocity field. We found that this numerical scheme accurately captured the average saturation, which could be seen by looking at the overall saturation profile and also by computing the fractional flow curves, which are a very important feature of interest to engineers when evaluating oil reservoir performance.

7.1 Further Work

The field of reservoir modeling is quite large and this thesis has addressed only a particular aspect of it. In particular, we assumed a specific model that ignored the effects of, for instance, cappillary pressure among other variables, and have focused our attention on the 2-D case. In modeling more realistic cases it is entirely possible that some of the problems faced here, such as the developement of sharp shocks in the solution for the saturation, would disappear. Nevertheless, within the current framework and numerical scheme there exist many potential avenues for improvement and further work. The most obvious of these is the development of a faster numerical scheme for the streamline projection. As mentioned in the section on numerical results, this was a major burden in the computations. Any improvements here could lead to a drastic cut in the run time for simulations. Also, coupling the scheme to an adaptive mesh refinement algorithm could also be useful. In this case, one can view the scheme as upscaling away from the oil-water front but also resolving features in this critical region.

An alternative to the scheme described could also be found by further modeling of the fluctuations so that instead of them being solved for explicitly their interaction with the large scales is modeled. Whilst this would possibly require a further set of assumptions to be made, the framework given provides an excellent means from which to do this.

Appendix A

Hyperbolicity of the Upscaled Saturation Equations

We need to demonstrate that the system (4.149), (4.150) is hyperbolic in the variables $\boldsymbol{y}$ and t. First, recall the definition of hyperbolicity for a system of equations [24]:

Definition A.1 *The quasilinear system*

$$\mathbf{q}_t + f'(\mathbf{q})\mathbf{q}_{x_1} + g'(\mathbf{q})\mathbf{q}_{x_2} = \mathbf{H}(x_1, x_2, t) \tag{A.1}$$

is (strongly) hyperbolic in some region of state space if the Jacobian matrix $\tilde{f}'(\mathbf{q}) = \mathbf{n}\cdot\mathbf{f}'(\mathbf{q}) = n_1 f'(\mathbf{q}) + n_2 g'(\mathbf{q})$ *is diagonalizable with real eigenvalues for every* $\mathbf{n} = (n_1, n_2)$*, for all* $\mathbf{q}$ *in this region.*

For our problem we take $\mathbf{q} = \left(\overline{S}, S'_{kl}\right)^T$, where $S'_{kl} = S'(\boldsymbol{x}, \boldsymbol{y}_{kl})$ identifies S' evaluated for the particular value of the parameter $\boldsymbol{y}_{kl}$. Notice that the right-hand side vector $\mathbf{H} = (0, G(\boldsymbol{x}, \boldsymbol{y}, t))^T$ can be evaluated *without* computing either $\frac{\partial S'}{\partial t}$ nor $\nabla_x S'$ since the terms $\mathcal{Q}\left(\nabla_x f'\right)$ and $\mathcal{Q}\left(\frac{\partial S'}{\partial t}\right)$ can be evaluated using the method given in Lemma 4.13. Therefore, we need not consider these in the evaluation of the Jacobian matrix and this is the reason why these are on the right-hand side. The system (4.150) can be written

$$\frac{\partial \mathbf{q}}{\partial t} + A\frac{\partial \mathbf{F}}{\partial x_1} + B\frac{\partial \mathbf{F}}{\partial x_2} = \mathbf{H} \tag{A.2}$$

where $\mathbf{F} = \left(\overline{f}, f'_{k,l}, \overline{\boldsymbol{v}'f'}\right)$

$$A = \begin{pmatrix} \overline{v}_1 & 0 & 1 \\ \mathcal{P}(v'_1) & \overline{v}_1 + \mathcal{P}(v'_1) & -1 \end{pmatrix} \tag{A.3}$$

$$B = \begin{pmatrix} \overline{v}_2 & 0 & 1 \\ \mathcal{P}(v'_2) & \overline{v}_2 + \mathcal{P}(v'_2) & -1 \end{pmatrix} \tag{A.4}$$

Thus, we need to evaluate the terms of the matrix

$$R = \begin{pmatrix} \frac{\partial \overline{f}}{\partial \overline{S}} & \frac{\partial \overline{f}}{\partial S'_{kl}} \\ \frac{\partial f'_{kl}}{\partial \overline{S}} & \frac{\partial f'_{kl}}{\partial S'_{kl}} \\ \frac{\partial \overline{\boldsymbol{v}'f'}}{\partial \overline{S}} & \frac{\partial \overline{\boldsymbol{v}'f'}}{\partial S'_{kl}} \end{pmatrix} \tag{A.5}$$

Recall that $\overline{f}$ and $\overline{\boldsymbol{v}'f'}$ are defined as the spatial averages in $\boldsymbol{y}$, i.e.

$$\overline{f}(\boldsymbol{x}, t) = \int_Y f(\overline{S} + S')d\boldsymbol{y} \tag{A.6}$$

$$\overline{\boldsymbol{v}'f'} = \int_Y \boldsymbol{v}'f'd\boldsymbol{y} \tag{A.7}$$

To derive expressions for (A.5) we write the integrals as Riemann sums. Consider the uniform partition $[y_{1,i-1}, y_{1,i}] \times [y_{2,j-1}, y_{2,j}]$ with center points $\boldsymbol{y}_{i,j}$ of the cell Y. Then, since we have assume that $\overline{S}$ and S' are bounded and continuous so that $f(\overline{S} + S')$ is bounded and continuous, we have

$$\sum_{i=1}^{I}\sum_{j=1}^{J} f(\overline{S} + S'_{i,j})\Delta y_{1,i}\Delta y_{2,j} \to \int_Y f(\overline{S} + S')d\boldsymbol{y} \tag{A.8}$$

as $I, J \to \infty$. Also, since $f' = f(\overline{S} + S') - \overline{f}$ we have

$$f(\overline{S} + S') - \sum_{i=1}^{I}\sum_{j=1}^{J} f(\overline{S} + S'_{i,j})\Delta y_{1,i}\Delta y_{2,j} \to f' \tag{A.9}$$

as $I, J \to \infty$. Now take derivatives directly using these: for brevity we write $\alpha_{i,j} = \Delta y_{1,i} \Delta y_{2,j}$

$$\frac{\partial \overline{f}}{\partial \overline{S}} = \sum_{i=1}^{I} \sum_{j=1}^{J} f_S(\overline{S} + S'_{i,j}) \alpha_{i,j} \tag{A.10}$$

$$\frac{\partial \overline{f}}{\partial S'_{k,l}} = f_S(\overline{S} + S'_{k,l}) \alpha_{k,l} \tag{A.11}$$

and

$$\frac{\partial f'_{k,l}}{\partial \overline{S}} = f_S(\overline{S} + S'_{k,l}) - \sum_{i=1}^{I} \sum_{j=1}^{J} f_S(\overline{S} + S'_{i,j}) \alpha_{i,j} \tag{A.12}$$

$$\frac{\partial f'_{k,l}}{\partial S'_{k,l}} = f_S(\overline{S} + S'_{k,l}) - f_S(\overline{S} + S'_{k,l}) \alpha_{k,l} \tag{A.13}$$

More generally, we have

$$\frac{\partial f'_{i,j}}{\partial S'_{k,l}} = \begin{cases} f_S(\overline{S} + S'_{k,l}) - f_S(\overline{S} + S'_{k,l}) \alpha_{k,l} & \text{if } i, j = k, l \\ -f_S(\overline{S} + S'_{k,l}) \alpha_{k,l} & \text{otherwise} \end{cases} \tag{A.14}$$

Thus, in the limit of $I, J \to \infty$ we have

$$\frac{\partial \overline{f}}{\partial \overline{S}} = \int_Y f_S(\overline{S} + S') d\boldsymbol{y} \tag{A.15}$$

$$\frac{\partial \overline{f}}{\partial S'_{k,l}} = 0 \tag{A.16}$$

$$\frac{\partial f'_{k,l}}{\partial \overline{S}} = f_S(\overline{S} + S'_{k,l}) - \int_Y f_S(\overline{S} + S') d\boldsymbol{y} \tag{A.17}$$

$$\frac{\partial f'_{k,l}}{\partial S'_{k,l}} = f_S(\overline{S} + S'_{k,l}) \tag{A.18}$$

Now consider the term $\overline{\boldsymbol{v}' f'}$. Writing this using the Riemann sum gives

$$\overline{\boldsymbol{v}' f'} = \int_Y \boldsymbol{v}' f' d\boldsymbol{y} \tag{A.19}$$

$$= \sum_{i=1}^{I} \sum_{j=1}^{J} \boldsymbol{v}'_{i,j} f'_{i,j} \alpha_{i,j} \tag{A.20}$$

Then

$$\frac{\partial\overline{\boldsymbol{v}'f'}}{\partial\overline{S}} = \frac{\partial\overline{\boldsymbol{v}'f'}}{\partial\overline{S}} \tag{A.21}$$

$$= \sum_{i=1}^{I}\sum_{j=1}^{J}\boldsymbol{v}'_{i,j}\frac{\partial f_{i,j}}{\partial\overline{S}}\alpha_{i,j} \tag{A.22}$$

$$= \sum_{i=1}^{I}\sum_{j=1}^{J}\boldsymbol{v}'_{i,j}f_S(\overline{S}+S'_{i,j})\alpha_{i,j} \tag{A.23}$$

where the first equality simply uses that fact that $\overline{\boldsymbol{v}'} = 0$ and we have also ignored the dependence of $\boldsymbol{v}'$ on S. Also,

$$\frac{\partial\overline{\boldsymbol{v}'f'}}{\partial S'_{k,l}} = \sum_{i=1}^{I}\sum_{j=1}^{J}\boldsymbol{v}'_{i,j}\frac{\partial f'_{i,j}}{\partial S'_{k,l}}\alpha_{i,j} \tag{A.24}$$

$$= \boldsymbol{v}'_{k,l}f_S(\overline{S}+S'_{k,l})\alpha_{k,l} - \sum_{i=1}^{I}\sum_{j=1}^{J}\boldsymbol{v}'_{i,j}f_S(\overline{S}+S'_{k,l})\alpha_{k,l}\alpha_{i,j} \tag{A.25}$$

$$= \boldsymbol{v}'_{k,l}f_S(\overline{S}+S'_{k,l})\alpha_{k,l} - f_S(\overline{S}+S'_{k,l})\alpha_{k,l}\sum_{i=1}^{I}\sum_{j=1}^{J}\boldsymbol{v}'_{i,j}\alpha_{i,j} \tag{A.26}$$

$$= \boldsymbol{v}'_{k,l}f_S(\overline{S}+S'_{k,l})\alpha_{k,l} \tag{A.27}$$

Thus, in the limit of $I, J \to \infty$ we have

$$\frac{\partial\overline{\boldsymbol{v}'f'}}{\partial\overline{S}} = \overline{\boldsymbol{v}'f_S} \tag{A.28}$$

$$\frac{\partial\overline{\boldsymbol{v}'f'}}{\partial S'_{k,l}} = 0 \tag{A.29}$$

Finally, the Jacobian matrices that we need, AR and BR are given by

$$AR = \begin{pmatrix} \overline{v}_1 & 0 & 1 \\ \mathcal{P}(v'_1) & \overline{v}_1+\mathcal{P}(v'_1) & -1 \end{pmatrix} \begin{pmatrix} \int_Y f_S(\overline{S}+S')d\boldsymbol{y} & 0 \\ f_S(\overline{S}+S'_{k,l}) - \int_Y f_S(\overline{S}+S')d\boldsymbol{y} & f_S(\overline{S}+S'_{k,l}) \\ \overline{\boldsymbol{v}'f_S} & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \overline{v}_1\overline{f_S} + \overline{v'_1f_S} & 0 \\ (\overline{v}_1+\mathcal{P}(v'_1))\,f_S - \left(\overline{v}_1\overline{f_S} + \overline{v'_1f_S}\right) & (\overline{v}_1+\mathcal{P}(v'_1))\,f_S \end{pmatrix} \tag{A.30}$$

with an exactly similar result for BR (with v_1 replaced by v_2). Then, consider the Jacobian $n_1AR + n_2BR$,

$$\begin{aligned} n_1AR + n_2BR &= n_1 \begin{pmatrix} \overline{v}_1\overline{f_S} + \overline{v_1' f_S} & 0 \\ \left(\overline{v}_1 + \mathcal{P}(v_1')\right) f_S - \left(\overline{v}_1\overline{f_S} + \overline{v_1' f_S}\right) & \left(\overline{v}_1 + \mathcal{P}(v_1')\right) f_S \end{pmatrix} \\ &+ n_2 \begin{pmatrix} \overline{v}_2\overline{f_S} + \overline{v_2' f_S} & 0 \\ \left(\overline{v}_2 + \mathcal{P}(v_2')\right) f_S - \left(\overline{v}_2\overline{f_S} + \overline{v_2' f_S}\right) & \left(\overline{v}_2 + \mathcal{P}(v_2')\right) f_S \end{pmatrix} \end{aligned} \tag{A.31}$$

By inspection, this has eigenvalues

$$\lambda_1 = n_1\left(\overline{v}_1\overline{f_S} + \overline{v_1' f_S}\right) + n_2\left(\overline{v}_1\overline{f_S} + \overline{v_2' f_S}\right) \tag{A.32}$$

$$\lambda_2 = n_1\left(\overline{v}_1 + \mathcal{P}(v_1')\right) f_S + n_2\left(\overline{v}_2 + \mathcal{P}(v_2')\right) f_S \tag{A.33}$$

In general these will be distinct and therefore the Jacobian is diagonalizable with real eigenvalues. Hence our system (4.149) and (4.150) is hyperbolic. ■

Appendix B

Special MSFEM Convergence

We consider here the convergence in the H^1 and L_2 norms only. The methods of proof are close to those given in [17] as the method is a nonconforming one. That the method is nonconforming can easily be seen from the construction of $\phi^i_{1,K}$ since, in general, $\frac{\partial \phi^i_0}{\partial x_j}$ will be discontinuous across cell boundaries.

Homogenization theory has been used in the analysis of MsFEM [20] and is useful in providing error estimates. The main results are briefly reviewed for completeness. It is know that the solution of (5.69) can be expanded as [3]

$$p_\epsilon = p_0 + \epsilon \chi^i \left(\frac{x}{\epsilon}\right) \frac{\partial p_0}{\partial x_i} + \epsilon \theta^p \tag{B.1}$$

Here $p_0 \in H^2(\Omega)$ is the solution of the homogenized equation

$$a^{ij}_* \frac{\partial^2}{\partial x_i x_j} p_0 = f, \text{ in } \Omega \tag{B.2}$$

satisfying $p_0 = 0$ on $\partial\Omega$. The constant homogenized coefficients a^{ij}_* are given by

$$a^{ik}_* = \frac{1}{|Y|} \int_Y a^{ij} \left(\delta_{jk} + \frac{\partial \chi k(y)}{\partial x_j}\right) dy \tag{B.3}$$

χ^k is the same as that given in (5.91) and the derivative of χ^k is with respect to the "fast" variable $y = x/\epsilon$.

For θ^p we have

$$-\nabla.\left(a(\frac{x}{\epsilon})\nabla\theta^p\right) = \frac{1}{\epsilon}\left(p_\epsilon - p_0 - \epsilon\chi^i\frac{\partial p_0}{\partial x_i}\right) \quad \text{(B.4)}$$

$$\theta^p|_{\partial\Omega} = -\chi_i\frac{\partial p_0}{\partial x_i} \quad \text{(B.5)}$$

B.1 H^1 estimates

Since the method is nonconforming, we cannot use Cea's lemma. The following lemma by Strang [30] gives the estimate of the error between the exact solution and the numerical solution in the energy norm

$$\|p_\epsilon - p_\epsilon^h\|_{h,\Omega} \leq C\left(\inf_{v_\epsilon^h\in\hat{H}^h}\|p_\epsilon - v_\epsilon^h\|_{h,\Omega} + \sup_{w_\epsilon^h\in\hat{H}^h}\frac{|f(w_\epsilon^h) - a(p_\epsilon, w_\epsilon^h)|}{\|w_\epsilon^h\|_h}\right) \quad \text{(B.6)}$$

where $\hat{H}^h$ is the finite dimensional space generated by the nonconforming basis functions (in general $\hat{H}^h \not\subset H^1$),

$$\|w_\epsilon^h\|_{h,\Omega} = \left(\sum_{K\in\mathbf{K}^h}\int_K |\nabla w_\epsilon^h|^2 dx)\right)^{1/2}, \quad \text{(B.7)}$$

and

$$\frac{|f(w_\epsilon^h) - a(p_\epsilon, w_\epsilon^h)|}{\|w_\epsilon^h\|_h} = \left|\int_\Omega f w_\epsilon^h dx - \sum_{K\in\mathbf{K}^h}\int_K a_\epsilon^{ij}\frac{\partial p_\epsilon}{\partial x_i}\frac{\partial w_\epsilon^h}{\partial x_j}dx\right| \quad \text{(B.8)}$$

We are considering the case in which $h \gg \epsilon$.

Theorem B.1.0.1 *Let p_ϵ be the solution of (5.69) and p_ϵ^h be the numerical solution computed using the MsFEM variant described. Assuming that p_0, the homogenized part of p_ϵ is in $W^{1,\infty}(\Omega)$, we have*

$$\|p_\epsilon - p_\epsilon^h\|_{h,\Omega} \leq C_1\sqrt{\epsilon} + C_2 h \quad \text{(B.9)}$$

Proof. The proof of this result is almost exactly the same as Theorem 3.1 in [17]. In that result, the only difference is that the θ^h terms (from the expansion of the basis functions) give a "resonant" term $C\epsilon/h$ in the estimate. Here, by construction of the basis functions, no such terms exist and hence there is no resonant term.

B.2 L_2 estimates

We use here a discrete error analysis [20, 17] to obtain the L_2 estimate. Denote the numerical solution of (B.2) using MsFEM by p_0^h. Since the coefficients a_*^{ij} are constants, the MsFEM method reduces to the standard linear finite element method and estimates from the stand theory apply. Therefore, we have

$$\begin{aligned} \|p_\epsilon - p_\epsilon^h\|_{L_2(\Omega)} &\leq \|p_\epsilon - p_0^h\|_{L_2(\Omega)} + \|p_0^h - p_\epsilon^h\|_{L_2(\Omega)} && \text{(B.10)} \\ &\leq C\epsilon + C_1 h^2 + C_2 \|p_0^h - p_\epsilon^h\|_{l_2(\Omega)} && \text{(B.11)} \end{aligned}$$

Let $N \sim 1/h^2$ be the number of nodal points. Denote $\|\cdot\|$ the standard maximum norm of matrices in $\mathbf{R}^{N\times N}$ and $|\cdot|$ the maximum norm of vectors in $\mathbf{R}^N$. The linear system of equations for P_ϵ^h is

$$A_\epsilon^h P_\epsilon^h = f_\epsilon^h \tag{B.12}$$

where A_ϵ^h and f_ϵ^h are obtained from $a(u^h, v)$ and $f(v)$ by using $v = \phi_\epsilon^i$ for $i = 1, \ldots, N$. Similarly, for P_0 one has

$$A_0^h P_0^h = f_0^h \tag{B.13}$$

where A_ϵ^h and f_ϵ^h are obtained by applying $v = \phi_0^i$, $i = 1, \ldots, N$ to $a^*(p_0^h, v) = f(v)$ with

$$a^*(p_0^h, v) = \int_\Omega a_*^{ij} v_{,i} p_{0,j}^h dx \tag{B.14}$$

The "comma" notation for partial differentiation is used here.

The main result is the following:

Theorem B.2.0.1 *Assuming that the distance between K and ∂S is of order h we have*

$$|P_\epsilon^h - P_0^h| \leq C_r \frac{\epsilon^2}{h^2} + C_1 \epsilon |\ln h| \quad (h \gg \epsilon) \tag{B.15}$$

The term $C_r \frac{\epsilon^2}{h^2}$ is referred to as the "cell resonance" in [17]. It follows from the above and (B.11) that for $h \gg \epsilon$

$$\|p_\epsilon - p_\epsilon^h\|_{L_2(\Omega)} \leq C\epsilon + C_1 h^2 + C_r \frac{\epsilon^2}{h^2} + C_2 \epsilon |\ln h| \tag{B.16}$$

Proof. Again the proof is almost exactly similar to the one given in [17]. P_ϵ^h has the following expansion

$$P_\epsilon^h = P_0^h + \sum_{i\geq 1} \epsilon^i P_i^h \tag{B.17}$$

where P_i^h $(i \geq 1)$ are given by

$$\begin{aligned} A_0^h P_1^h &= f_1^h - A_1^h P_0^h & \text{(B.18)} \\ A_0^h P_i^h &= -A_1^h P_{i-1}^h \quad (i > 1) & \text{(B.19)} \end{aligned}$$

There is no contribution from "boundary layers" in the basis functions due to there construction here. Therefore, in the (c.f. [17]) we have

$$\begin{aligned} A_\epsilon^h &= A_0^h + \epsilon A_1^h & \text{(B.20)} \\ f_\epsilon^h &= f_0^h + \epsilon f_1^h & \text{(B.21)} \end{aligned}$$

where A_ϵ^h and f_ϵ^h are as described above and

$$\begin{aligned} A_1^h &= \Lambda_2^h & \text{(B.22)} \\ f_1^h &= F_2^h & \text{(B.23)} \end{aligned}$$

(see equation 4.18 in [17]). Explicitly,

$$\begin{aligned} \Lambda_{2_{kl}}^h &= \frac{1}{\epsilon}\int_K \tilde{\sigma}^{ij}\phi_{0,j}^l \phi_{0,i}^k dx & \text{(B.24)} \\ F_2^h &= -\int_K f(x)\chi^p \phi_{0,p}^i dx & \text{(B.25)} \end{aligned}$$

Both Λ_2^h and F_2^h have a "difference structure", i.e.

$$\begin{aligned} \Lambda_2^h &= D\lambda^h & \text{(B.26)} \\ F_2^h &= D\hat{f}^h & \text{(B.27)} \end{aligned}$$

The analysis is exactly the same as in [17] except that there are no terms involving the

boundary correction θ^h. Therefore, we get the result (corresponding to 4.26 in [17]) that

$$|\epsilon P_1^h| \leq C\epsilon |\ln h| + C_1\epsilon \tag{B.28}$$

and

$$|\epsilon^2 P_2^h| \leq C\frac{\epsilon^2}{h^2} \tag{B.29}$$

Further terms in the expansion for P_ϵ^h are of higher order. Substituting the above into this expansion gives (B.15).

B.3 Numerical results

In this section we investigate numerically the above estimates (B.15) and (B.16). We first describe some of the implementation details for the scheme. We implement the scheme using a rectangular mesh coarse mesh of size $M \times N$, so that the coarse mesh sizes is $h_x = 1/M$ and $h_y = 1/N$. In fact, in our tests we always take $M = N$ and so we write $h_x = h$. This gives our partition of the domain Ω into the set of rectangular elements $\mathbf{K}^h$. Within each of these elements K, we solve for the basis functions given by

$$\phi_K^i = \phi_{0,K}^i + \epsilon\phi_{1,K}^i \tag{B.30}$$

where $\phi_{0,K}^i$ is the standard linear basis function in the element K with $\phi_{0,K}^i(x_j) = \delta_{ij}$ and $\phi_{1,K}^i$ is given

$$\phi_{1,K}^i = -\chi^j \frac{\partial \phi_{0,K}^i}{\partial x_j} \tag{B.31}$$

where χ^j is the periodic solution of

$$\nabla_y . \left(a(y)\nabla_y \chi^j\right) = \frac{\partial}{\partial y_i} a_{ij}(y) \tag{B.32}$$

in a unit cell Y with zero mean, i.e.

$$\langle \chi^j \rangle = \frac{1}{|Y|}\int_Y \chi^j dy = 0 \tag{B.33}$$

χ^j is solved for once at the start of the computations. We solve for χ^j using a linear finite element method with 512×512 grid points. This gives χ^j extremely accurately. To evaluate χ^j at non-grid points (which is needed since in general h/ϵ will not be an integer) we use bilinear interpolation. Since this computation is performed only once at the start, it is not absolutely necessary to have a very fast solver for this equation. However, here we use a black-box multigrid that is capable of handling the perioidic boundary conditions [11]. We evaluate each basis function at $K \times L$ points within each element, choosing $K = L$ and also $K = 4096/M$. This is done so that the fine grid reconstruction always has 4096×4096 grid points.

Once the basis functions have been computed, we need to compute the stiffness matrix. This is done by integrating overlapping elements. The integration is done using the trapeziodal rule over the fine grid points. The right-hand side of the linear system is evaluated in a likewise manner.

The resulting linear system is solved using a multigrid solver, [33]. This gives the homogenized solution P_0^h at coarse-grid points. The multiscale solution P_ϵ^h is then constructed by interpolation of this coarse-grid solution using the multiscale basis functions, i.e.

$$P_\epsilon^h = \sum_{m,n} P_{0,mn}^h \phi_{mn} \tag{B.34}$$

where $P_{0,mn}^h$ is the coarse-grid solution at the grid point m, n and ϕ_{mn} is the basis function that is 1 at m, n, zero at all other coarse-grid points.

In the first set of experiments, we numerically investigate the error estimate (B.15), which is crucial in getting the error estimate for the L_2 norm of the error in the multiscale solution. We perform the same basic tests as were done in [17]. That is, we fix the ratio $\alpha = h/\epsilon$ and then decrease the mesh size h and compute the l_2 norm numerically. The purpose of these experiments in [17] was to show that the "resonance" error, due to the term of the form $C\epsilon/h$ in the l_2 error estimate, was indeed observable. By decreasing h and ϵ simulataneously with their ratio fixed, this resonant term should remain and this was observed for the case of $\alpha = 1.5$.

In this new method, we do not have such a term, but we still have the term $C_r\epsilon^2/h^2$. In [17] it was reported that this error was generically small and very difficult to observe. Our numerical experiments also show this to be true for the new method, i.e. that the constant

h	$\alpha = 1.5$		$\alpha = 1.5625$		$\alpha = \sqrt{10}/2 \approx 1.58$	
	l_2	rate	l_2	rate	l_2	rate
1/16	6.302e^{-4}		6.010e^{-4}		5.800e^{-4}	
1/32	2.906e^{-4}	1.117	2.864e^{-4}	1.069	2.791e^{-4}	0.965
1/64	1.461e^{-4}	0.992	1.426e^{-4}	1.006	1.444e^{-4}	1.041
1/128	7.109e^{-5}	1.039	7.083e^{-5}	1.010	7.174e^{-5}	1.009
1/256	3.510e^{-5}	1.018	3.496e^{-5}	1.019	3.549e^{-5}	1.015
1/512	1.661e^{-5}	1.079	1.698e^{-5}	1.041	1.724e^{-5}	1.042
1/1024	6.7922e^{-6}	1.290	7.952e^{-6}	1.094	8.138e^{-6}	1.083

Table B.1: $||P_\epsilon^h - P_0^h||_{l_2}$ for various $\alpha = h/\epsilon$.

h	$\alpha = 1.0$		$\alpha = 1.2$		$\alpha = 1.9$	
	l_2	rate	l_2	rate	l_2	rate
1/16	3.331e^{-4}		4.108e^{-4}		7.014e^{-4}	
1/32	1.657e^{-4}	0.999	2.101e^{-4}	0.967	3.800e^{-4}	0.884
1/64	8.275e^{-5}	1.002	1.017e^{-4}	0.932	1.950e^{-4}	0.963
1/128	4.117e^{-5}	1.007	5.084e^{-5}	1.116	9.267e^{-5}	1.073
1/256	2.017e^{-5}	1.030	2.487e^{-5}	1.032	4.617e^{-5}	1.006
1/512	9.139e^{-6}	1.142	1.775e^{-5}	0.486	2.236e^{-5}	1.046
1/1024	2.948e^{-6}	1.632	5.043e^{-6}	1.816	1.085e^{-5}	1.043

Table B.2: $||P_\epsilon^h - P_0^h||_{l_2}$ for various $\alpha = h/\epsilon$ that were not considered in [17].

C_r in (B.15) is extremely small and difficult to observe. Table B.1 summarizes the results.

We see that for $\alpha = 1.5$, the rate remains close to 1 and does not stagnate, as was found when using the standard MSFEM in [17]. For that case, it was shown, by use of Green's functions, that the resonance error should be greatest for the case of $\alpha = 1.5$. Since we have not performed such an analysis for the term $C_r\epsilon^2/h^2$ we also checked some different values of α which are further away from 1.5, results that were not shown in [17]. For these cases, we still see that the convergence rate remains close to 1. Other tests were performed with different choices for the function a in () and similar result were obtained. Therefore, we conclude that the constant C_r in (B.15) must be extremely small, or in fact there exists further error cancellations so that it is an overestimate of the true error.

We also compute some l_2 errors for the solution P_ϵ^h we compared with "exact" solutions, to verify the error bound (B.16). By exact, we mean very well resolved computations, carried out using the standard linear finite element method with a 2048×2048 grid (which was the maximum number we were able to handle on our current computing resources). For

h	l_2	rate
1/8	5.691e^{-4}	
1/16	3.963e^{-4}	0.522
1/32	3.401e^{-5}	0.221
1/64	3.187e^{-5}	0.094

Table B.3: $\|p_\epsilon - p_\epsilon^h\|_{L_2(\Omega)}$ computed for various MSFEM grids.

these tests we use $\epsilon = \sqrt{2}/100 \approx 0.0070710$ so that the resolved computations have about 14 grid points in each direction per period. This value of ϵ is not as small as we would like, but as mentioned above, we were limited by hardware resources (in particular memory, which becomes an issue for the multigrid solver used). We compute the MSFEM solution p_ϵ^h using $M \times M$ grids, with $M = 8, 16, 32$, with sub-cell resolution $K = 2048/M$ so that we have 2048×2048 fine grid points for the reconstructed solution. Table B.3 summarizes the results. We see that error quickly stagnates due to the fact that even for $M = 16$ we have $h^2 \approx 0.00390625$ which is smaller than ϵ. The convergence rate between 8 and 16 clearly indicates, however, that the scheme is indeed convergent. A more satisfactory test would involve using a smaller value of ϵ. This would allow us to see the second order convergence in h of the error $\|p_\epsilon - p_\epsilon^h\|_{L_2(\Omega)}$.

Appendix C

Implementation of the Hyperbolic Solver via CLAWPACK

We have the system

$$\frac{\partial \mathbf{Q}}{\partial t} + A\frac{\partial \mathbf{Q}}{\partial x_1} + B\frac{\partial \mathbf{Q}}{\partial x_2} = 0 \tag{C.1}$$

where A and B are matrices, with

$$A = \begin{pmatrix} \overline{v}_1 & 0 & \cdots & \cdots & \cdots \\ \mathcal{P}(v_1')_{0,0} & \overline{v}_1 + \mathcal{P}(v_1')_{0,0} & 0 & \cdots & \cdots \\ \mathcal{P}(v_1')_{0,1} & 0 & \overline{v}_1 + \mathcal{P}(v_1')_{0,1} & \cdots & \cdots \\ \vdots & & \ddots & & \end{pmatrix} \tag{C.2}$$

In CLAWPACK, a wave propagation approach is used to calculate fluxes. This involves computing a set of eigenvectors of the matrices A. From the structure of A we can immediately see that $\mathbf{e}_i$ $i = 2, \ldots, 1 + KL$ are eigenvectors, with eigenvalues $\lambda_i = \overline{v}_1 + \mathcal{P}(v_1')_{k,l}$. Here $\mathbf{e}_i$ denotes with usual unit vectors in $\mathbb{R}^{1+KL}$ with elements $(\mathbf{e}_i)_j = \delta_{ij}$. Another eigenvector $\mathbf{u}$ is determined easily by setting the eigenvalue $\lambda_1 = A_{11} = \overline{v}_1$. Then, $u_1 = 1$, and the other elements are computed using

$$A_{n1} + A_{nn}u_n = \lambda u_n = A_{11}u_n \Rightarrow u_n = \frac{A_{n1}}{A_{11} - A_{nn}} \tag{C.3}$$

and then adjusting this to make the length 1. Thus, we have a set of eigenvectors.

We then need to express the flux difference $\Delta\mathbf{q}$ as a combination of these eigenvectors.

We have

$$\Delta\mathbf{q} = \sum_{i=1}^{1+KL} \beta_i \mathbf{u}_i \tag{C.4}$$

The set $\{\mathbf{u}_i\}$, $i = 2, \ldots, 1 + KL$ is orthonormal. However, $\mathbf{u}_1$ is not orthogonal to vectors in this set. To calculate these β_i, note that

$$\begin{aligned} \Delta\mathbf{q} \cdot \mathbf{u}_1 &= \beta_1 + \sum_{i=2}^{1+KL} \beta_i (\mathbf{u}_i \cdot \mathbf{u}_1) && \text{(C.5)} \\ \Delta\mathbf{q} \cdot \mathbf{u}_k &= \beta_1 (\mathbf{u}_1 \cdot \mathbf{u}_k) + \beta_k, \ k = 2, \ldots 1 + KL && \text{(C.6)} \end{aligned}$$

So we get the linear system

$$\begin{pmatrix} 1 & \mathbf{u}_1 \cdot \mathbf{u}_2 & \mathbf{u}_1 \cdot \mathbf{u}_3 & \ldots & \ldots \\ \mathbf{u}_1 \cdot \mathbf{u}_2 & 1 & 0 & \ldots & \ldots \\ \mathbf{u}_1 \cdot \mathbf{u}_3 & 0 & 1 & 0 & \ldots \\ \vdots & & \ddots & & \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} \Delta\mathbf{q} \cdot \mathbf{u}_1 \\ \Delta\mathbf{q} \cdot \mathbf{u}_2 \\ \Delta\mathbf{q} \cdot \mathbf{u}_3 \\ \vdots \end{pmatrix} \tag{C.7}$$

From this we see, we row manipulation that

$$\begin{aligned} \beta_1 &= \frac{\Delta\mathbf{q} \cdot \mathbf{u}_1 - \sum_{i=2}^{1+KL} (\mathbf{u}_i \cdot \mathbf{u}_1) \Delta\mathbf{q} \cdot \mathbf{u}_i}{1 - \sum_{i=2}^{1+KL} (\mathbf{u}_i \cdot \mathbf{u}_1)^2} && \text{(C.8)} \\ &= \frac{\Delta\mathbf{q} \cdot \mathbf{u}_1 - \sum_{i=2}^{1+KL} u_{1,i} \Delta\mathbf{q} \cdot \mathbf{u}_i}{1 - \sum_{i=2}^{1+KL} u_{1,i}^2} && \text{(C.9)} \end{aligned}$$

Then,

$$\beta_k = \Delta\mathbf{q} \cdot \mathbf{u}_k - u_{1,k} \beta_1 \tag{C.10}$$

With these, we then have

$$A\Delta\mathbf{q} = \sum_i \beta_i \lambda_i \mathbf{u}_i \tag{C.11}$$

Bibliography

[1] D. Bale, R. LeVeque, S. Mitran, and J. Rossmanith. A wave propagation method for conservation laws and balance laws with spatially-varying flux functions. *SIAM J. Sci. Comput.*, 2002.

[2] R. Beckie, A. A. Aldama, and E. F. Wood. Modeling the large-scale dynamics of saturated groundwater flow using spatial filtering theory, 1, theoretical development. *Water Resour. Res.*, 32:1269–1280, 1996.

[3] A. Bensoussan, J. Lions, and G. Papanicolaou. *Asymptotic analysis for periodic structures*, volume 5 of *Studies in Mathematics and Its Applications*. North-Holland Publ., 1978.

[4] M. Blunt, L. Liu, and M. Thiele. A generalized streamline method to predict reservoir flow. *Petroleum geoscience*, 02:259–269, 1996.

[5] D. Braess. *Finite Elements. Theory, Fast Solvers and Applications in Solid Mechanics.* Cambridge University Press, 2001.

[6] S. E. Buckley and M. C. Leverett. Mechanisms of fluid displacements in sands. *Trans. AIME*, 146:107–116, 1942.

[7] H. Ceniceros and T. Y. Hou. An efficient dynamically adaptive mesh for potentially singular solutions. *J. Comput. Phys.*, 172:1–31, 2001.

[8] Z. Chen and T. Y. Hou. A mixed multiscale finite element method for elliptic problems with oscillating coefficients. *Math. Comput.*, 2000.

[9] K. H. Coats. A note on impes and some impes-based models. *SPE Journal*, 5(3):245–251, 2000.

[10] P. Concus, G. Golub, and G. Meurant. Block preconditioning for the conjugate gradient method. *SIAM J. Sci. Stat. Comput.*, 6(1), 1985.

[11] J. Dendy. Multigrid for periodic and singular problems. *Applied Math. and Comp.*, 25:1–10, 1988.

[12] C. Deutsch and A. G. Journel. *GSLIB: Geostatistical Software Library and User's Guide*. Oxford Univ. Press, New York, 2nd edition, 1998.

[13] L. J. Durlofsky. Numerical calculation of equivalent grid block permeability tensors for hetrogeneous porous media. *Water Resour. Res.*, 27:699–708, 1991.

[14] Y. Efendiev and L. J. Durlofsky. Numerical modeling of subgrid hetrogeneity in two phase flow simulations. *Water Resour. Res.*, 2002.

[15] Y. Efendiev, L. J. Durlofsky, and S. H. Lee. Modeling of subgrid effects in coarse-scale simulations of transport in hetrogeneous porous media. *Water Resour. Res.*, 36(8):2031–2041, 2000.

[16] Y. R. Efendiev. *The Multiscale Finite Element Method and its Applications.* PhD thesis, Caltech, 1999.

[17] Y. R. Efendiev, T. Y. Hou, and X. H. Wu. Convergence of a nonconformal multiscale finite element method. *SIAM J. Numer. Anal.*, 2000.

[18] I. G. Kevrekidis et al. Coarse integration/bifurcation analysis via microscopic simulators: micro-galerkin methods. *Computers and Chemical Engineering*, submitted.

[19] T. Y. Hou and X. H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[20] T. Y. Hou, X. H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Math. Comput.*, 68:913–943, 1999.

[21] P. Jenny, S. Lee, and H. Tchelepi. Multi-scale finite-volume method for elliptic problems in subsurface flow simulation. *Journal of Computational Physics*, to appear.

[22] P. Langlo and M. Espedal. Macrodispersion for two-phase, immiscible flow in porous media. *Advances in Water Resources*, 17:297–316, 1994.

[23] R. LeVeque. *CLAWPACK 4.1 User's Guide.*

[24] R. LeVeque. *Finite-Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[25] A. Oberman. Convergent difference schemes for degenerate elliptic and parabolic equations: Hamilton-jacobi equations and free boundary problems. *SINUM*, 2003.

[26] O. A. Oleinik. Uniqueness and stability of the generalized solution of the cauchy problem for a quasilinear equation. *Amer. Math Soc. Transl.*, 33:285–290, 1964.

[27] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21:217–235, 1984.

[28] D. Peaceman. *Fundamentals of Numerical Reservoir Simulation.* Elsevier Scientific Publishing Co., 1977.

[29] H. L. Stone and A. Garder Jr. Analysis of gas-cap or dissolved-gas drive reservoirs. *Trans. AIME*, 222, 1961.

[30] G. Strang and G. Fix. *An anlysis of the finite element method.* Prentice-Hall, Englewood Cliffs, NJ, 1973.

[31] M. Thiele, R. Batycky, M. Blunt, and R. Orr. Simulating flow in hetrogeneous systems using streamtubes and streamlines. *SPE Reservoir Engineering*, 11:5–12, 1996.

[32] G. B. Whitham. *Linear and nonlinear waves.* Wiley, New York, 1974. 636 p.

[33] P. M. De Zeeuw. Matrix-dependent prolongation and restrictions in a blackbox multigrid solver. *J. Comput. Phys.*, 33:1–27, 1990.