



UNIVERSITY OF
CAMBRIDGE

MPhil Dissertation

**The Grid Orientation Effect In Miscible
Displacement**

Peter Wirnsberger
Selwyn College

August 2012

Supervisors:

Dr N. Nikiforakis
Department of Physics
University of Cambridge

Dr F. Monmont
Schlumberger Cambridge Research

This dissertation is submitted for the degree of Master of Philosophy.

Abstract

Many numerical methods show a dependence of the solution on the orientation of the underlying computational grid with respect to the flow. Initially small numerical errors can have considerable effects on the result when the model exhibits physical instabilities. This effect is known as the grid orientation effect (GOE). Extensive research has been carried out on the GOE for miscible displacement, and similar effects are also observed for the numerical simulation of multi-phase, multi-component thermal flow in porous media. This investigation aims to find a numerical scheme that minimizes the GOE for miscible displacement whilst being suitable for extension to the more complex case of thermal flow in porous media. We present a novel approach of partially stabilizing the concentration front by employing a level set function.

Preface

The work contained in this thesis has been carried out between October 2011 and August 2012, in the Cavendish Laboratory, Department of Physics, University of Cambridge. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

Peter Wirsberger, August 2012

Acknowledgements

I want to thank Dr Nikos Nikiforakis for supervising the project, and Schlumberger Cambridge Research (SCR) for providing my funding. Furthermore I want to thank Dr Franck Monmont at SCR for fruitful discussions. Especially I want to thank Sean Lovett and Dr Stephen Millmore for invaluable assistance throughout the whole project. Finally I want to thank my parents for all their support.

Contents

1	Introduction	1
1.1	Enhanced oil recovery processes	1
1.2	Motivation	1
1.3	Previous work	3
1.4	Outline	4
2	Miscible displacement	5
2.1	Governing equations	5
2.1.1	Simplifying assumptions	6
2.1.2	Adverse displacement	7
2.2	Quarter five-spot problem	8
2.3	Well model	10
3	Numerical scheme	12
3.1	Staggered grid	13
3.2	Domain boundary	13
3.3	Elliptic solver	14
3.3.1	Transformation to the rotated coordinate system	16
3.3.2	Discretization of the pressure equation	17
3.3.3	Incorporation of Neumann boundary conditions	20
3.4	Calculation of the velocity	23
3.5	Level set solver	24
3.5.1	Level set methods	25
3.5.2	Level set scheme	26
3.6	Hyperbolic solver	27
3.6.1	Splitting scheme	28
3.6.2	Advection equation	30
3.6.3	Source term	36

3.6.4	Parabolic equation	37
3.6.5	Maximum stable time step	39
3.6.6	Convergence tests	39
4	Numerical results	43
4.1	Parameters	44
4.2	Demonstration of the grid orientation effect	45
4.3	Comparison for CTU and MUSCL	45
4.3.1	No numerical dispersion	47
4.3.2	Numerical dispersion of Shubin and Bell	49
4.3.3	Modified numerical dispersion	53
4.3.4	Errors for mass and velocity	53
4.4	Partial dispersion study	58
5	Conclusions	63
5.1	Future work	64
	Bibliography	65

List of Figures

2.1	Well pattern in the quarter five-spot configuration	9
2.2	Projection of the parallel solution	10
3.1	Flowchart of the IMPES method	12
3.2	Staggered grid	14
3.3	Weighted stencil for the pressure equation	15
3.4	Coordinate transformation	16
3.5	Illustration of the nine-point stencil approximation	19
3.6	Incorporation of Neumann boundary conditions	21
3.7	Calculation of the velocity	24
3.8	Implicit representation of a circle	25
3.9	Second-order splitting scheme	29

3.10	First-order splitting scheme	30
3.11	Wave propagation formulation of the CTU scheme	36
3.12	Convergence test for hyperbolic solver	41
3.13	Convergence test for elliptic solver	42
4.1	Demonstration of the GOE	47
4.2	CTU without dispersion	48
4.3	MUSCL without dispersion	50
4.4	CTU with numerical dispersion of Shubin and Bell	51
4.5	MUSCL with numerical dispersion of Shubin and Bell	52
4.6	CTU with new numerical dispersion	54
4.7	MUSCL with new numerical dispersion	56
4.8	Mass error for uniform dispersion	57
4.9	Error for the divergence of the velocity	57
4.10	MUSCL with partial dispersion	59
4.11	Contour plots for uniform and partial dispersion	60
4.12	Slices from the injection well to the production well	61
4.13	Mass error for partial dispersion	62

List of Tables

4.1	Simulation parameters	46
-----	---------------------------------	----

Introduction

Optimizing the process of recovering oil from a reservoir is of great importance for the petroleum industry. Mathematical formulations of various physical models have been devised in order to predict the performance of particular recovery schemes. These models usually consist of coupled, non-linear partial differential equations that, in general, cannot be solved analytically. Therefore computational simulations help us to better understand the physical and chemical phenomena associated with reservoir flows [12].

1.1 ENHANCED OIL RECOVERY PROCESSES

The oil is usually trapped in microscopic pores in the rock, and high pressure gradients are necessary to force it to flow. Different techniques are utilized to produce the oil. An important class of recovery techniques are *Enhanced Oil Recovery* (EOR) processes.

One type of EOR is the injection of a chemical species such as CO_2 , which will mix with the hydrocarbon and undergo a phase change. This type of replacement procedure is termed *miscible displacement* [12].

Another, more complex, thermal process is *in-situ combustion*. In this approach, an ignition due to the injection of oxygen, and possibly a fluid, leads to the formation of a self-sustaining, propagating combustion front which increases the mobility of the oil and allows for production [24, 9]. This method is particularly interesting for the exploitation of heavy-oil reservoirs [9].

1.2 MOTIVATION

Recent work of van Odyck *et al.* [33, 34] provides a mathematical formulation, as well as numerical solutions, of multiphase, multicomponent thermal flow in porous media.

Such systems are challenging to solve numerically for several reasons. The governing equations are coupled and non-linear, and the various physical phenomena happen on length scales differing by several orders of magnitude [12]. Furthermore, for each time step expensive phase equilibrium ("flash") calculations are required in order to determine the phase decomposition of the species. Highly efficient algorithms are therefore desirable [34].

Another aspect which increases the complexity of solving the governing flow equations numerically is physical instabilities at the modelling scale [17]. In order to address this problem, we impose simplifying assumptions on the equations considered in [33]. They then reduce to miscible displacement, which serves as an easier, well studied problem that has been investigated extensively in the past five decades [5]. Carrying out stability analysis for radial source flow, Tan and Homsy [30] found that the displacement is destabilized by two parameters, the mobility ratio and the Péclet number. Difficulties in the simulation arise since the numerical errors can bias or even trigger such instabilities [17]. In some cases this results in incorrect numerical results as observed by Todd *et al.* [31]. They found that different orientations of the underlying grid lead to completely different solutions, which is called the *grid orientation effect* (GOE). A famous test problem to illustrate this effect is the so-called *quarter five-spot problem*, which will be explained in detail in the next chapter.

Since its discovery, the GOE has been studied extensively due to its importance for reservoir simulations. Many new methods aiming to eliminate the problem have been devised. These improved methods are successful in that the more advanced schemes nearly eliminate grid orientation for a certain range of resolutions [2, 6, 18]. However, these methods are typically complicated and hence computationally expensive. For complex systems like thermal flow in porous media, this additional computational cost might be substantial, since reconstruction of states, for example, involves carrying out additional phase equilibrium calculations. This important aspect motivates the present investigation of the GOE.

The aim of this report is to compare the grid orientation effect exhibited by various explicit finite volume schemes that are suitable for the simulation of thermal flow in porous media. A first-order rotationally invariant numerical dispersion tensor is employed for stabilization. This suppresses a physical instability for the sake of numerical consistency between different grid orientations. Throughout the report, the term 'dispersion' refers to velocity-dependent mixing unless explicitly stated otherwise. Compared to previous work, a wider range of resolutions are investigated in this report. In order to track the evolution of the concentration front, a level set function is employed and a novel approach of

partially stabilizing the front is presented. If this is to be a successful approach, it should minimize the effect of grid orientation whilst not being too expensive, in order for it to be applicable to thermal flow simulations.

1.3 PREVIOUS WORK

Previous approaches concerning the GOE range from employing methods that better capture the directionality of the flow [36, 16, 27, 28, 11] and adding stabilizing rotationally invariant numerical diffusion-dispersion terms [29, 10, 2, 13, 6, 18], to adapting the grid itself [19]. A good summary of past work is given in Kozdon [17].

Comparisons to past work are not easy, because most of the available results are low-resolution (less than 100×100 cells), apart from a few exceptions [8, 5, 37]. In some cases this was probably due to insufficient computational power, however, even in more recent papers, e.g. [18, 19], only low-resolution results of less than 100×100 cells are provided. Since the GOE does depend on the grid spacing, as demonstrated by Kozdon [17] for the method devised by Shubin and Bell [29], this is disappointing.

Furthermore, the treatment of the injection and production wells in the quarter five-spot configuration is important. Depending on the underlying method, modelling the wells as a point source can lead to very large pressure gradients in the vicinity of the wells [12]. Therefore, some authors use special procedures for incorporating the wells, e.g. [29, 5]. Also, the stability of the model is determined by two parameters and the various schemes perform differently depending on the choice of these parameters. In this investigation, we try to choose the parameters in such a way that the results are comparable to previous investigations and we provide high-resolution results for the two symmetric configurations.

Finally, the previous work ranges from employing fully implicit methods to explicit methods. The former are more stable, but require solving large systems of equations for each time step, whilst the latter have better computational features but suffer from more restrictive stability criteria. A variety of methods, termed IMPES, solve implicitly for pressure and explicitly for saturation. These methods are widely used, but not particularly efficient due to the small time steps needed for stability [12]. However, in this report only methods that follow the IMPES approach are considered, because this approach is used in simulations of thermal flow in porous media [34].

Accurate modelling of physical phenomena such as fingering is beyond the scope of this report. Readers who are interested in this are referred to [7, 5]. The main interest lies

in minimizing the difference in the solutions of the diagonal and parallel grid in the quarter five-spot configuration. As a means of achieving this, dispersion must be added, which is not necessarily physical, but serves in stabilizing the numerical solution. In fact, by adding high enough amounts of numerical diffusion fingering can be completely suppressed. Hence it is suggested that numerical diffusion and dispersion are kept to a minimum [5].

1.4 OUTLINE

In Chapter 2, the equations for miscible displacement are derived, emphasizing the relation to thermal flow in porous media. The quarter five-spot problem is also explained in this chapter. The numerical schemes for solving this problem are presented in Chapter 3. Separate convergence tests for the elliptic and hyperbolic solvers are presented to support their correct implementation. The results are finally presented in Chapter 4, qualitatively comparing the performance of various hyperbolic schemes. In the last chapter, the findings are summarized and possible future improvements are addressed.

Miscible displacement

As outlined in the introduction there is a relationship between the equations modelling multi-phase, multi-component thermal flow in porous media and the simpler set of equations used in this report. In the following section we will derive the equations for miscible displacement. We begin the derivation from the mass conservation laws and then consider the effects of diffusion and dispersion.

2.1 GOVERNING EQUATIONS

The flow in a porous medium is governed by a set of equations that attempt to track the behaviour of the fluid [17]. The model for miscible displacement does not take into account temperature and comprises only a simple one-phase, two-component flow. However in the absence of diffusion-dispersion terms, we can derive the formulation of these equations from a more general system of conservation laws. Following [17], we have a number of chemical components, n_c , for which we have a vector of mass concentrations, $\mathbf{c} = (c_1, \dots, c_{n_c})^T$, in a multi-phase, multi-component flow. These obey a set of conservation laws of the form

$$\phi \frac{\partial c_i}{\partial t} + \nabla \cdot \mathbf{F}_i(\mathbf{c}) = q_i \quad (2.1.1)$$

where ϕ models the porosity of the medium, $\mathbf{q} = (q_1, \dots, q_{n_c})^T$ represents the vector of source/sink terms and $\mathbf{F}_i(\mathbf{c})$ is the mass-flux vector for component i . The components are allowed to flow together in n_p different phases and α denotes the phase index. The saturation S_α of phase α , is the volume fraction of the fluid in this phase. Therefore we require

$$\sum_{\alpha=1}^{n_p} S_\alpha = 1 \quad (2.1.2)$$

to be satisfied. We can also define the fraction of component i in phase α as $x_{i,\alpha}$. The vector of mole fractions in phase α is then given by [33]

$$\mathbf{x}_\alpha = (x_{1,\alpha}, \dots, x_{n_c,\alpha})^\top = \frac{\mathbf{n}_\alpha}{\mathbf{e}^\top \mathbf{n}_\alpha} \quad (2.1.3)$$

where \mathbf{n}_α is the vector of moles of each component in phase α per pore volume and $\mathbf{e} = (1, 1, \dots, 1)^\top \in \mathbb{R}^{n_c}$. We can easily see that

$$\sum_{i=1}^{n_c} x_{i,\alpha} = \mathbf{e}^\top \mathbf{x}_\alpha = \frac{\mathbf{e}^\top \mathbf{n}_\alpha}{\mathbf{e}^\top \mathbf{n}_\alpha} = 1. \quad (2.1.4)$$

The relation between these quantities and the phase densities, ρ_α , is given by [17]

$$\mathbf{c} = \sum_{\alpha=1}^{n_p} \rho_\alpha \mathbf{x}_\alpha S_\alpha. \quad (2.1.5)$$

Neglecting gravity and capillary pressure relationships, the average velocity of a fluid phase α can be written as

$$\mathbf{u}_\alpha = -\frac{k_{r,\alpha}}{\mu_\alpha} \mathbf{K} \nabla p \quad (2.1.6)$$

where \mathbf{K} is the absolute permeability tensor, $k_{r,\alpha}$ the relative permeability and μ_α the viscosity of phase α . Whereas \mathbf{K} is a property of the rock defining how easily the fluid can flow, the relative permeability $k_{r,\alpha}$ defines how easily an individual phase can flow in the presence of others. Finally, the mass-flux can be written as [17]

$$\mathbf{F}_i = \sum_{\alpha=1}^{n_p} x_{i,\alpha} \rho_\alpha \mathbf{u}_\alpha. \quad (2.1.7)$$

These equations define the flow in thermodynamic phase equilibrium [17].

2.1.1 Simplifying assumptions

We now impose the simplifying assumptions of incompressible flow (density is constant in space and time) of just two chemical components c_1 and c_2 . Both chemicals are assumed to be in a single phase, thus $n_p = 1$. If we multiply equation (2.1.1) by \mathbf{e}^\top from the left, we get

$$\phi \frac{\partial(c_1 + c_2)}{\partial t} + \nabla \cdot (\mathbf{F}_1 + \mathbf{F}_2) = q_1 + q_2 \quad (2.1.8)$$

and from the equations (2.1.2) and (2.1.5) it follows, that the time derivative of the first term vanishes. Using the equations (2.1.4) and (2.1.7), we can further simplify equation (2.1.8) to

$$\nabla \cdot (\mathbf{u} \rho) = q_1 + q_2 \quad (2.1.9)$$

where $\mathbf{u} := \mathbf{u}_1$ and $\rho := \rho_1$. Furthermore, if we define $c := c_1$, $\tilde{c} := c_2$, $q_1 := \hat{c}q$ and $q_2 := \tilde{c}q$, then the equations (2.1.1) and (2.1.9) become

$$\phi \frac{\partial}{\partial t} \begin{pmatrix} c \\ \tilde{c} \end{pmatrix} + \begin{pmatrix} \nabla \cdot (x_{1,1} \mathbf{u} \rho) \\ \nabla \cdot (x_{2,1} \mathbf{u} \rho) \end{pmatrix} = q \begin{pmatrix} \hat{c} \\ \tilde{c} \end{pmatrix} \quad (2.1.10)$$

$$\nabla \cdot (\rho \mathbf{u}) = q \quad (2.1.11)$$

given that \hat{c} and \tilde{c} add up to unity. From equation (2.1.5) it follows that $c = x_{1,1}\rho$ and if we set ρ to be unity, we get

$$\phi \frac{\partial}{\partial t} \begin{pmatrix} c \\ 1-c \end{pmatrix} + \begin{pmatrix} \nabla \cdot (c \mathbf{u}) \\ \nabla \cdot ((1-c) \mathbf{u}) \end{pmatrix} = q \begin{pmatrix} \hat{c} \\ 1-\hat{c} \end{pmatrix} \quad (2.1.12)$$

$$\nabla \cdot \mathbf{u} = q \quad (2.1.13)$$

$$\mathbf{u} = -\frac{k_{r,1}}{\mu} \mathbf{K} \nabla p \quad (2.1.14)$$

since we can replace \tilde{c} by $1-c$. Therefore, it is only necessary to track the concentration of one fluid. Additionally, the factor $\frac{k_{r,1}}{\mu}$ is replaced by an effective mobility of the form [29]

$$\frac{1}{\mu(c)} = \left(\frac{c}{\mu_c^{\frac{1}{4}}} + \frac{1-c}{\mu_{\tilde{c}}^{\frac{1}{4}}} \right)^4 \quad (2.1.15)$$

where μ_c and $\mu_{\tilde{c}}$ are the viscosities of the two fluids respectively. Adding a diffusion-dispersion term, this finally allows us to rewrite the set of equations as presented in [29]

$$\phi \frac{\partial}{\partial t} c + \nabla \cdot (c \mathbf{u}) - \nabla \cdot \mathbf{D} \nabla c = q \hat{c} \quad (x, y, t) \in \Omega \times (0, T) \quad (2.1.16)$$

$$\nabla \cdot \mathbf{u} = q \quad (x, y) \in \Omega \quad (2.1.17)$$

$$\mathbf{u} = -\frac{\mathbf{K}}{\mu(c)} \nabla p \quad (x, y, t) \in \Omega \times (0, T) \quad (2.1.18)$$

in two spatial dimensions. In this report, fluid is always injected with concentration equal to unity and the production wells produce whatever fluid happens to be present. Therefore, we can express \hat{c} as [29]

$$\hat{c} = \begin{cases} 1 & \text{if } q > 0, \\ c & \text{if } q < 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2.1.19)$$

2.1.2 Adverse displacement

The equations for miscible displacement exhibit instabilities in two dimensionless parameters, as presented in [30] for radial source flow. The Péclet number, Pe , relates source

strength to diffusivity and affects the dynamics of the flow, i.e. the viscous fingering process [5]. A definition of Pe is given in §3.6.4, where we introduce the numerical dispersion tensor. The second parameter is the mobility ratio M , which is the ratio of the viscosities of the resident fluid (μ_r) and the injected fluid (μ_c). If diffusion and dispersion are neglected and M is greater than unity, the displacement is unstable. In this case the displacement process is called *adverse* [29].

As detailed in [6], physical diffusion and dispersion terms are mathematically necessary for stabilizing the set of equations (2.1.16)–(2.1.18). However, physical effects cannot always be resolved by the numerical scheme unless the resolution is very high. Therefore, it is necessary to replace \mathbf{D} by a numerical diffusion-dispersion term to provide stabilization.

2.2 QUARTER FIVE-SPOT PROBLEM

As mentioned previously, the GOE is triggered by numerical errors due to non-rotationally invariant operators. In order to demonstrate the effect, at least two different grid orientations with respect to the flow have to be considered. The *quarter five-spot* configuration illustrated in Fig. 2.1 consists of a periodically repeated pattern of production and injection wells. Choosing a square grid with equal mesh spacing in both dimensions such that the vertices coincide with an injection and a production well will cause flow diagonal to the grid lines. Fixing one vertex, rotating the grid by 45° and upscaling it by $\sqrt{2}$ results in a second configuration, where the flow is parallel to the grid lines. The two configurations will be referred to as *diagonal* and *parallel* grid, respectively.

The positions of the wells are chosen such that the symmetry in Fig. 2.1 holds. In this representation every four adjacent wells act as a point source which is four times as strong and located at the symmetry point. However, in the finite volume discretization they are represented as cell averages. Another well configuration considered in this report models the wells as circles with a cut-off radius (see §2.3).

For the solutions to be comparable, we choose the same grid spacing for both grid types. Since an edge of the parallel grid is $\sqrt{2}$ times longer than the corresponding diagonal one, the resolution for the parallel grid is taken to be $\sqrt{2}$ times higher. Once the solution for the parallel grid (*parallel solution*) is calculated, we project a quarter of the parallel domain onto the upper part of the diagonal domain to obtain the diagonal solution, as illustrated in Fig. 2.2.

Let \mathbf{x}^D denote the coordinates of a diagonal grid cell centre with respect to the dashed

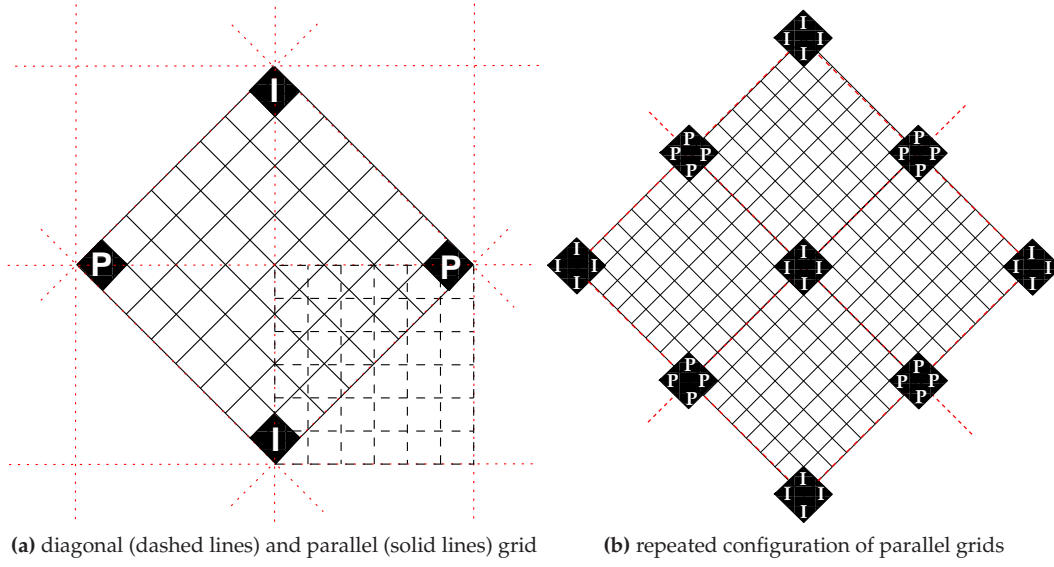


Figure 2.1: Repeated well pattern for the quarter five-spot configuration, where I marks an injection well and P a production well. (a) The solid black line is the parallel grid, the black dashed line constitutes the diagonal grid and the red dotted line indicates no-flow boundaries. (b) A repeated configuration for the parallel grid consisting of four grids.

base in Fig. 2.2. Then, for each cell of the diagonal grid, the bilinear interpolation is carried out in the following four steps:

1. Given \mathbf{x}^D (red point), calculate the coordinates \mathbf{x}^P with respect to the base for the parallel grid (solid base in Fig. 2.2).
2. If the y-component of \mathbf{x}^P is negative, change its sign according to the symmetry conditions.
3. Determine the cell centres of the four nearest neighbouring parallel grid cells (blue points) with respect to \mathbf{x}^P .
4. Numerate the parallel grid points according to $\mathbf{x}_i = (x_i, y_i)$, $i = 1, \dots, 4$, clockwise starting with the one closest to the chosen origin. Use the same numeration scheme for the average parallel grid cell values q_i . The interpolated value for the diagonal

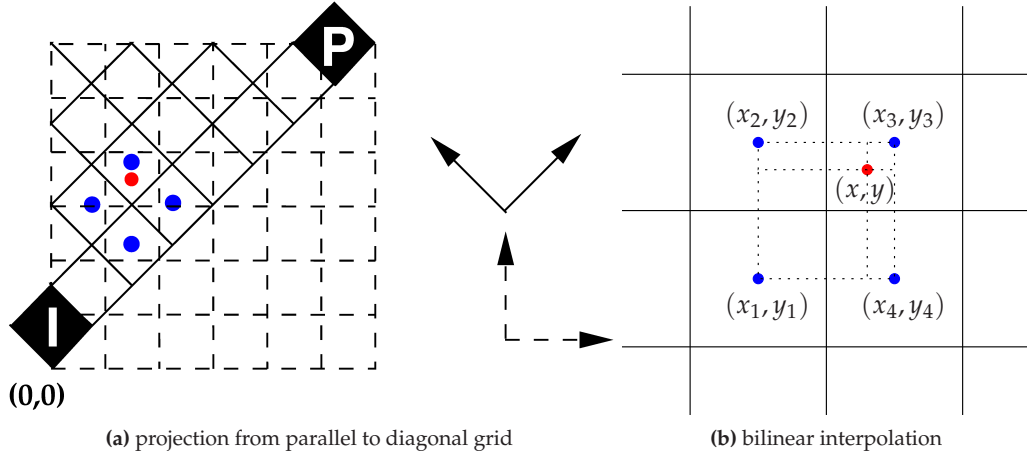


Figure 2.2: Four-point stencil for the bilinear projection of the parallel solution (blue points) onto the diagonal one (red point).

cell is then given by

$$\frac{1}{\Delta x \Delta y} \left[(x_3 - x)(y_3 - y)q_1 + (x_4 - x)(y - y_4)q_2 + (x - x_1)(y - y_1)q_3 + (x - x_2)(y_2 - y)q_4 \right].$$

From this formula it is easy to see that the contribution of each of the parallel grid values is weighted by the area spanned by \mathbf{x}^P and the respective diagonally opposite parallel grid point (see dashed lines in Fig. 2.2b).

Employing this interpolation procedure, the diagonal grid solution obtained from the parallel one is by definition symmetric. Asymmetries of the parallel solution are therefore eliminated in the projection step. This is justified because all the employed operators are symmetric with respect to the spatial directions, even in the cases where dimensional splitting is applied (see §3.6.1).

2.3 WELL MODEL

In this report we consider two different well models: point sources and circular wells. The source terms for both representations are denoted by $q^P(x, y)$ and $q^C(x, y)$ and their discretized versions by \mathbf{q}^P and \mathbf{q}^C , respectively. Independently of the model, we want

to inject the same amount of fluid per unit time for comparison. We can achieve this by requiring that the integral averages are the same. For the injection well in Fig. 2.2a, this is satisfied, if the relation

$$\int_0^{\Delta x} dx \int_0^{\Delta y} dy q^P(x, y) = \int_0^{r_C} r dr \int_0^{\frac{\pi}{2}} d\phi \tilde{q}^C(r, \phi) \quad (2.3.1)$$

holds, where r_C denotes the cut-off radius and $\tilde{q}^C(r, \phi) := q^C(x, y)$ describes the circular well in polar coordinates. By symmetry, relation (2.3.1) holds also for the second well in the diagonal configuration and for the three other wells in the parallel one. The functions $q^P(x, y)$ and $\tilde{q}^C(x, y)$ are step functions, with constant values q^P and q^C inside the well and zero outside, respectively. Carrying out the integration, we get

$$q^P \Delta x \Delta y = \frac{q^C r_C^2 \pi}{4}, \quad (2.3.2)$$

which relates both models. In the case of a circular well, exact volume fractions are taken into account when \mathbf{q}^C is calculated.

Numerical scheme

Van Odyck *et al.* used explicit schemes in their recent studies [33, 34]. We follow this strategy for solving the non-linear, elliptic-hyperbolic set of equations (2.1.16)-(2.1.18) and use the IMPES approach explained in Fig. 3.1.

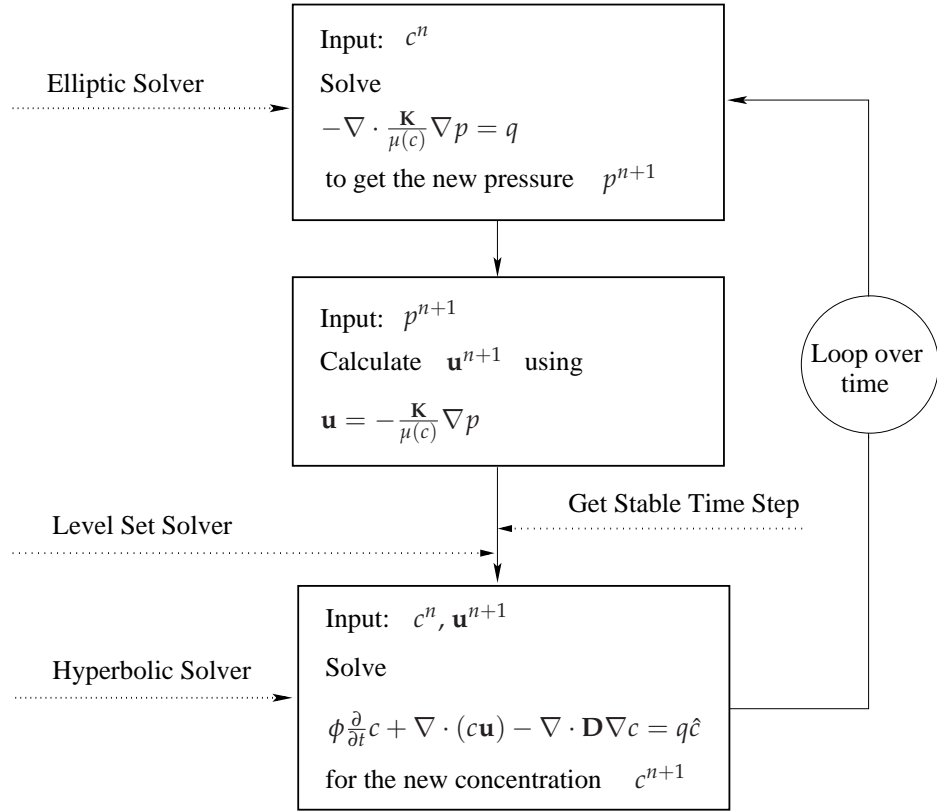


Figure 3.1: IMPES method.

For each time step, the elliptic equation (2.1.17) is solved implicitly to obtain the pressure for the next time step. Subsequently, the velocity is calculated from equation (2.1.18) to convect the concentration field. This approach completely decouples the elliptic and hyperbolic solvers and allows different schemes to be combined with ease. In this chapter, we will first present the staggered grid that is employed. The benefit of this choice will become apparent in the following sections when the schemes are explained in detail. The implementation of both solvers is supported by individual convergence tests against analytical solutions. Finally, the operator splitting and incorporation of the dispersion term is presented along with a stability analysis to determine the maximum stable time step.

3.1 STAGGERED GRID

There are a variety of different grids available, some of which are more suitable for incompressible flow problems than others [23]. Most of the past work on the GOE outlined in the introduction has been carried out using structured grids. However, some recent work has also investigated structured and unstructured quadrilateral grids [19]. In this report we use a structured, staggered grid (see Fig. 3.2), as used in [34], for the following reasons:

- it allows Neumann boundary conditions and equation (2.1.17) to be satisfied simultaneously and straightforwardly;
- the pressure-velocity decoupling observed for the incompressible Navier-Stokes equations does not occur [23];
- the velocities do not coincide with the wells, when they are modelled as point sources at the cell centre;
- many hyperbolic schemes require velocities at cell interfaces so that no interpolation is needed.

3.2 DOMAIN BOUNDARY

In order to update a specific cell, the numerical scheme requires information about the neighbouring cells. Close to the boundary of the computational domain, this information is not always available. However, we can add additional layers of cells called *ghost cells* (shaded area in Fig. 3.2a) and use them whenever information from outside the domain is needed. The question of how to define the states of these ghost cells remains. For

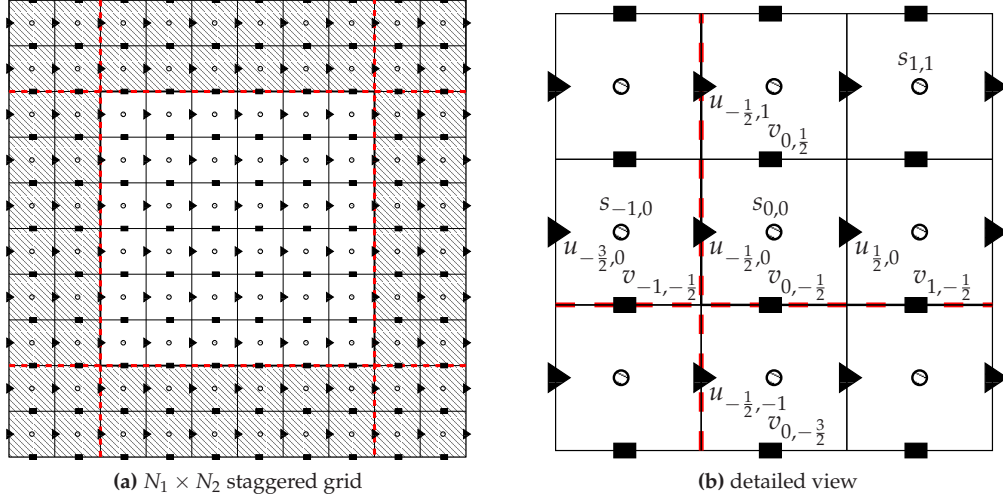


Figure 3.2: Staggered grid; (a) full grid with two layers of ghost cells; the red dashed lines represent no-flow boundaries, scalar quantities (pressure, concentration and level set function) are stored at the cell centre and denoted by circles (s). Black triangles (u) and rectangles (v) represent the velocity $\mathbf{u} = (u, v)^T$; (b) detailed view of the left, bottom corner.

the quarter-five spot configuration, we want to keep the symmetry of the periodically repeated well configuration. This can be achieved by imposing *no-flow* boundary conditions [29]. These boundaries are such that the gradient of scalar quantities like pressure, concentration or the level set function (denoted by s in Fig. 3.2b) vanish at the domain boundary. Let us consider the pressure in ghost cell $(-1, 0)$. The value $p_{-1,0}$ is set to $p_{0,0}$, yielding a zero pressure gradient at the boundary when central differences are applied. The boundaries are such that the velocity is reflected ($u_{-\frac{3}{2},0} = -u_{\frac{1}{2},0}$ and $v_{0,-\frac{3}{2}} = -v_{0,\frac{1}{2}}$) implying a vanishing normal velocity at the boundary.

3.3 ELLIPTIC SOLVER

In this section, we first transform the pressure equation (2.1.17) and then discretize it. The computational domain Ω is divided into a grid of N_1 by N_2 cells, as shown in Fig. 3.2a. We want to estimate the $M = N_1 \times N_2$ values

$$p((l + 0.5)\Delta x^1, (m + 0.5)\Delta x^2), \quad 0 \leq l \leq N_1 - 1, \quad 0 \leq m \leq N_2 - 1, \quad (3.3.1)$$

numerically. For this, we want to write the discretized form of equation (2.1.17) as a linear system of equations

$$\mathbf{A}\mathbf{p} = -\mathbf{q} + \mathbf{b} \quad (3.3.2)$$

where $\mathbf{A} \in \mathbb{R}^{M \times M}$ and \mathbf{p} represents the pressure stored at the cell centres in a column-major ordering. The right-hand side of equation (3.3.2) consists of two parts: \mathbf{q} is the discretized form of the source term and \mathbf{b} incorporates correction terms for the domain boundary. The explicit form of \mathbf{A} depends on the stencil we choose. Yanosik and McCracken [36] realized the importance of employing a stencil that takes into account all eight neighbouring cells in order to minimize the GOE. In their work, they combined two five-point stencils rotated by 45° with respect to each other (see Fig. 3.3a).

We follow a similar but more general approach. As Yanosik and McCracken only allow for uniform mobility, the mobility tensor does not change under transformation to rotated coordinates [36]. This means central difference approximations of the spatial derivatives in the rotated coordinate system again lead to a five-point stencil. In the more general case of a diagonal permeability tensor, the same approximation leads to a nine-point stencil in the rotated coordinate system. Thus, we combine a five-point stencil contribution arising from the unrotated coordinate system with a nine-point stencil contribution from the rotated coordinate system (see Fig. 3.3b).

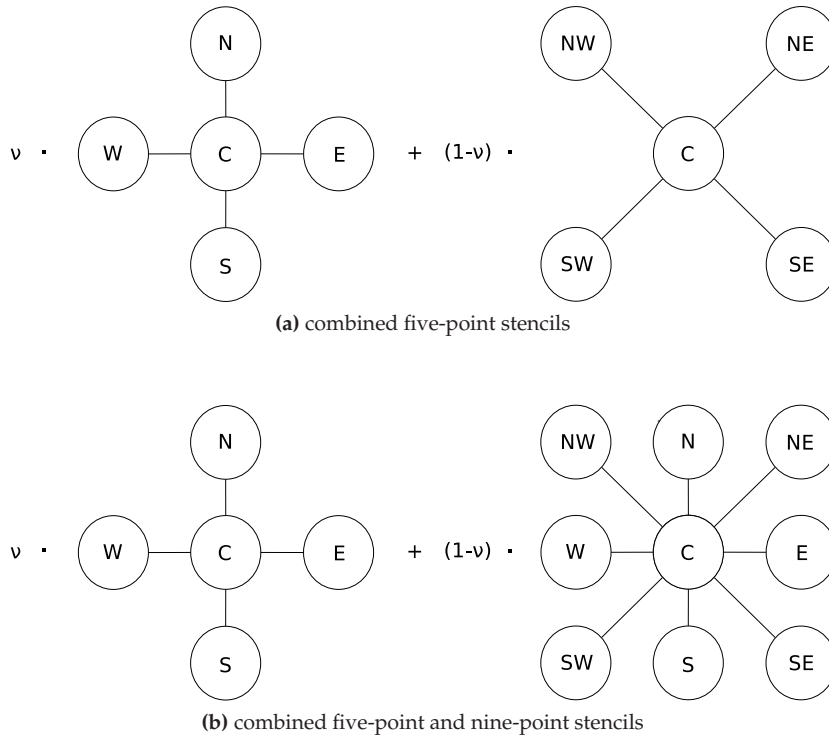


Figure 3.3: (a) Weighted nine-point stencil consisting of a *parallel* stencil (left) and a *diagonal* stencil (right) as used by Yanosik and McCracken [36]. (b) Weighted nine-point stencil consisting of a five-point stencil (unrotated coordinates) and a nine-point stencil (rotated coordinates) contribution.

Under the assumption of a diagonal permeability tensor, we can cast equations (2.1.17)–(2.1.18) into

$$\nabla \cdot \begin{pmatrix} a(x^1, x^2) & 0 \\ 0 & \tilde{a}(x^1, x^2) \end{pmatrix} \nabla p(x^1, x^2) = -q(x^1, x^2) \quad (3.3.3)$$

with $\mathbf{M} := \text{diag}(a, \tilde{a}) = \mathbf{K}\mu(c)^{-1}$. We will formulate the discretization in terms of the mobility tensor \mathbf{M} . Application of the five-point stencil is straightforward in the (x^1, x^2) -coordinate system, since it is chosen to coincide with our grid axes. However, care must be taken when employing the diagonal stencil in the rotated coordinate system because the mobility tensor needs to be transformed as well.

3.3.1 Transformation to the rotated coordinate system

We will assume equal grid spacing in both spatial dimensions throughout this section. In this case, the diagonal stencil is rotated by 45° with respect to the parallel one, which simplifies the transformation and makes the equations more readable. The extension to rectangular grid cells is straightforward. The new coordinates (ξ^1, ξ^2) are chosen such that the original basis vectors \mathbf{e}_1 and \mathbf{e}_2 are rotated by 45° , illustrated in Fig. 3.4.

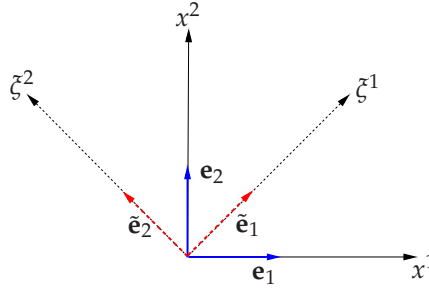


Figure 3.4: Rotation of the basis vectors.

The new basis vectors are then given by

$$\begin{pmatrix} \tilde{\mathbf{e}}_1 \\ \tilde{\mathbf{e}}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{pmatrix} \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}, \quad (3.3.4)$$

where

$$\mathbf{R} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{R}^{-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \quad (3.3.5)$$

are rotation matrices. The coordinates transform according to

$$\begin{pmatrix} \xi^1 \\ \xi^2 \end{pmatrix} = \mathbf{R}^{-1} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} \xi^1 \\ \xi^2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} x^1 + x^2 \\ x^2 - x^1 \end{pmatrix}$$

and

$$\begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \xi^1 - \xi^2 \\ \xi^1 + \xi^2 \end{pmatrix}. \quad (3.3.6)$$

Using Einstein summation convention, the derivatives and infinitesimal changes transform as

$$\tilde{\partial}_l = R_l^m \partial_m \quad \Leftrightarrow \quad \tilde{\partial}_1 = \frac{1}{\sqrt{2}}(\partial_1 + \partial_2) \quad \text{and} \quad \tilde{\partial}_2 = \frac{1}{\sqrt{2}}(\partial_2 - \partial_1) \quad (3.3.7)$$

$$d\xi^i = (\mathbf{R}^{-1})^i_j dx^j \quad \Leftrightarrow \quad d\xi^1 = \frac{1}{\sqrt{2}}(dx^1 + dx^2) \quad \text{and} \quad d\xi^2 = \frac{1}{\sqrt{2}}(dx^2 - dx^1) \quad (3.3.8)$$

where

$$R_l^m = \frac{\partial x^m}{\partial \xi^l} \quad \text{and} \quad (\mathbf{R}^{-1})^i_j = \frac{\partial \xi^i}{\partial x^j}. \quad (3.3.9)$$

Since \mathbf{R} is a rotation, the metric tensor does not change and we have $g_{ij} = \tilde{g}_{ij} = \delta_{ij}$. We can rewrite equation (3.3.3) in the new coordinate system as

$$g^{ij} \partial_i M_j^l \partial_l p = \partial_i M^{ij} \partial_j p = -q \quad \longrightarrow \quad \tilde{\partial}_i \tilde{M}^{ij} \tilde{\partial}_j \tilde{p} = -\tilde{q}, \quad (3.3.10)$$

where $\tilde{p}(\xi^i) := p(x^j(\xi^i))$ and $\tilde{q}(\xi^i) := q(x^j(\xi^i))$. The tensor transforms according to

$$\tilde{M}^{ij} = (\mathbf{R}^{-1})^i_k (\mathbf{R}^{-1})^j_l M^{kl} \quad \Leftrightarrow \quad \tilde{\mathbf{M}} = \mathbf{R}^{-1} \mathbf{M} \mathbf{R} = \frac{1}{2} \begin{pmatrix} \tilde{a} + a & \tilde{a} - a \\ \tilde{a} - a & \tilde{a} + a \end{pmatrix} \quad (3.3.11)$$

where a and \tilde{a} are now functions of (ξ^1, ξ^2) . Finally, we can cast equation (3.3.3) into

$$\boxed{\tilde{\partial}_1 \tilde{M}^{11} \tilde{\partial}_1 \tilde{p} + \tilde{\partial}_1 \tilde{M}^{12} \tilde{\partial}_2 \tilde{p} + \tilde{\partial}_2 \tilde{M}^{21} \tilde{\partial}_1 \tilde{p} + \tilde{\partial}_2 \tilde{M}^{22} \tilde{\partial}_2 \tilde{p} = -\tilde{q}} \quad (3.3.12)$$

together with the definitions made in equation (3.3.11) and equation (3.3.7).

3.3.2 Discretization of the pressure equation

For both coordinate systems, the pressure equation has the same form as equation (3.3.10). However, the derivatives and tensors are defined with respect to the underlying coordinate system. We only present the discretization in the (ξ^1, ξ^2) -coordinate system, as it is the more general case and from which the other follows easily. We use central difference operators to approximate the spatial derivatives at the cell centres:

$$(\tilde{\partial}_1 \tilde{p})_{ij} = \frac{\tilde{p}_{i+\frac{1}{2}j} - \tilde{p}_{i-\frac{1}{2}j}}{\Delta \xi} + \mathcal{O}(\Delta \xi^2) \quad (3.3.13)$$

$$(\tilde{\partial}_2 \tilde{p})_{ij} = \frac{\tilde{p}_{ij+\frac{1}{2}} - \tilde{p}_{ij-\frac{1}{2}}}{\Delta \xi} + \mathcal{O}(\Delta \xi^2), \quad (3.3.14)$$

where $\Delta\zeta = \sqrt{2}\Delta x$, and the mesh spacing is the same in both dimensions. Applying these operators to the second term in equation (3.3.12) yields

$$\tilde{\partial}_1 \tilde{M}^{12} \tilde{\partial}_2 \tilde{p} \simeq \frac{1}{\Delta\zeta} \left[\tilde{M}_{i+\frac{1}{2}j}^{12} \frac{1}{\Delta\zeta} (\tilde{p}_{i+\frac{1}{2}j+\frac{1}{2}} - \tilde{p}_{i+\frac{1}{2}j-\frac{1}{2}}) - \tilde{M}_{i-\frac{1}{2}j}^{12} \frac{1}{\Delta\zeta} (\tilde{p}_{i-\frac{1}{2}j+\frac{1}{2}} - \tilde{p}_{i-\frac{1}{2}j-\frac{1}{2}}) \right].$$

Continuing analogously for the other terms, leads, after reordering, to

$$\begin{aligned} & \tilde{p}_{i+1j} \tilde{M}_{i+\frac{1}{2}j}^{11} + \tilde{p}_{i-1j} \tilde{M}_{i-\frac{1}{2}j}^{11} + \tilde{p}_{ij+1} \tilde{M}_{ij+\frac{1}{2}}^{22} + \tilde{p}_{ij-1} \tilde{M}_{ij-\frac{1}{2}}^{22} \\ & - \tilde{p}_{ij} \left(\tilde{M}_{i+\frac{1}{2}j}^{11} + \tilde{M}_{i-\frac{1}{2}j}^{11} + \tilde{M}_{ij+\frac{1}{2}}^{22} + \tilde{M}_{ij-\frac{1}{2}}^{22} \right) + \tilde{p}_{i+\frac{1}{2}j+\frac{1}{2}} \left(\tilde{M}_{i+\frac{1}{2}j}^{12} + \tilde{M}_{ij+\frac{1}{2}}^{21} \right) \\ & - \tilde{p}_{i-\frac{1}{2}j+\frac{1}{2}} \left(\tilde{M}_{ij+\frac{1}{2}}^{21} + \tilde{M}_{i-\frac{1}{2}j}^{12} \right) - \tilde{p}_{i+\frac{1}{2}j-\frac{1}{2}} \left(\tilde{M}_{i+\frac{1}{2}j}^{12} + \tilde{M}_{ij-\frac{1}{2}}^{21} \right) \\ & + \tilde{p}_{i-\frac{1}{2}j-\frac{1}{2}} \left(\tilde{M}_{i-\frac{1}{2}j}^{12} + \tilde{M}_{ij-\frac{1}{2}}^{21} \right) + \mathcal{O}(\Delta\zeta^2) = -\tilde{q}_{ij} \Delta\zeta^2. \end{aligned} \quad (3.3.15)$$

The final step is to associate \tilde{p}_{kl} with p_{ij} , and to define the nine coefficients of the stencil. Although the half-indices in equation (3.3.15) suggest that we need interpolated pressure values at the vertices, only cell centred values are required, as illustrated in Fig. 3.5. If we were to apply the same approximation to a full tensor in (x^1, x^2) -coordinates, we would need values at the vertices. This will become apparent when discretizing the dispersion term in §3.6. However, we need the components of the mobility tensor at the vertices. Therefore, we calculate the harmonic averages of the functions a and \tilde{a} at the cell vertices and build the linear combinations defined in equation (3.3.11).

The order of the discretization error is the same in both coordinate systems, since $\Delta x^2 \propto \Delta\zeta^2$. However, the spacing is bigger in the rotated coordinate system, which has an influence on our choice of the weighting factor ν in Fig. 3.3b. From Fig. 3.5, we can see that the relations

$$\begin{aligned} \tilde{p}_{i+1j} &= p_{i+1j+1}, & \tilde{p}_{i-1j} &= p_{i-1j-1}, & \tilde{p}_{ij+1} &= p_{i-1j+1}, \\ \tilde{p}_{i-\frac{1}{2}j-\frac{1}{2}} &= p_{ij-1}, & \tilde{p}_{i+\frac{1}{2}j-\frac{1}{2}} &= p_{i+1j}, & \tilde{p}_{i+\frac{1}{2}j+\frac{1}{2}} &= p_{ij+1}, \\ \tilde{p}_{i-\frac{1}{2}j+\frac{1}{2}} &= p_{i-1j}, & \tilde{p}_{ij-1} &= p_{i+1j-1}, & \tilde{p}_{ij} &= p_{ij}, \end{aligned}$$

hold. Although we omitted the indices for the nine coefficients \mathcal{NW} , \mathcal{N} , \mathcal{NE} , \mathcal{W} , \mathcal{C} , \mathcal{E} , \mathcal{SW} , \mathcal{S} and \mathcal{SE} in Fig. 3.5, those quantities vary in space. Thus, for every cell (i, j) in $\{0, \dots, N_1 - 1\} \times \{0, \dots, N_2 - 1\}$, we have to recalculate the coefficients that define the linearized equation for that cell. If all eight neighbours are available, this equation can be written as

$$\begin{aligned} \mathcal{NW}_{ij} p_{i-1j+1} + \mathcal{N}_{ij} p_{ij+1} + \mathcal{NE}_{ij} p_{i+1j+1} + \mathcal{W}_{ij} p_{i-1j} + \mathcal{C}_{ij} p_{ij} + \mathcal{E}_{ij} p_{i+1j} \\ + \mathcal{SW}_{ij} p_{i-1j-1} + \mathcal{S}_{ij} p_{ij-1} + \mathcal{SE}_{ij} p_{i+1j-1} = -q_{ij} + b_{ij} \end{aligned} \quad (3.3.16)$$

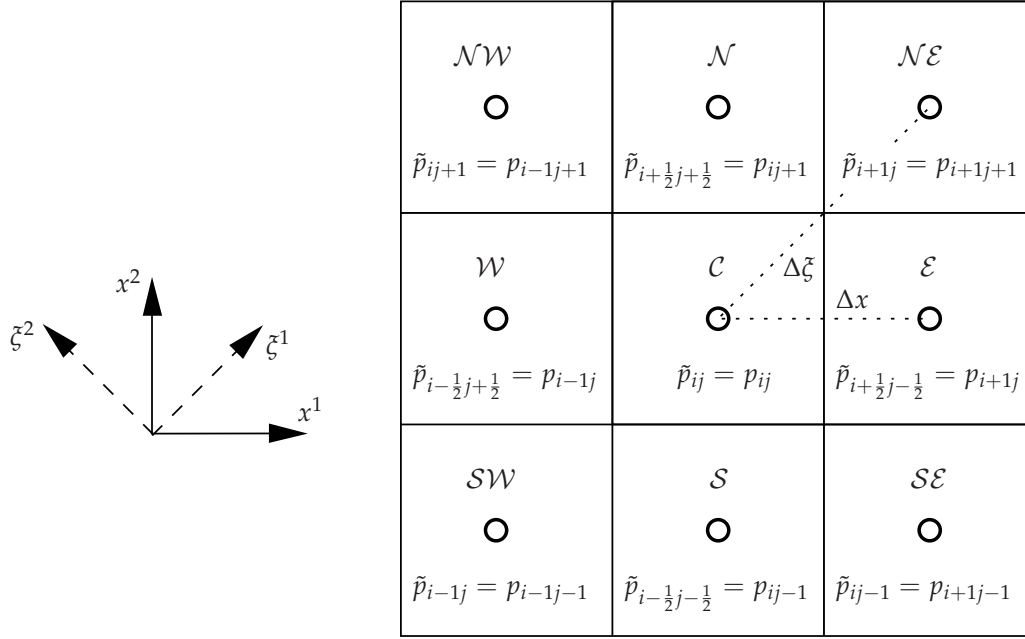


Figure 3.5: Illustration of the approximation given by equation (3.3.15).

with $b_{ij} = 0$, i.e. no boundary corrections are necessary. However, for the cell layer for which the edges coincide with the boundary, the stencil requires points outside of the domain, which are missing. Specification of either pressure values (Dirichlet conditions) or gradients (Neumann conditions) on the domain boundary enable us to express the missing values as linear functions of the available points inside the domain. This will be explained in detail in §3.3.3. The coefficients in equation (3.3.16) are given by

$$\begin{aligned}
 \mathcal{NW}_{ij} &= \frac{(\tilde{a} + a)_{ij+\frac{1}{2}}}{2\Delta\tilde{\xi}^2} \\
 \mathcal{N}_{ij} &= \frac{(\tilde{a} - a)_{i+\frac{1}{2}j} + (\tilde{a} - a)_{ij+\frac{1}{2}}}{2\Delta\tilde{\xi}^2} \\
 \mathcal{NE}_{ij} &= \frac{(\tilde{a} + a)_{i+\frac{1}{2}j}}{2\Delta\tilde{\xi}^2} \\
 \mathcal{W}_{ij} &= -\frac{(\tilde{a} - a)_{ij+\frac{1}{2}} + (\tilde{a} - a)_{i-\frac{1}{2}j}}{2\Delta\tilde{\xi}^2} \\
 \mathcal{C}_{ij} &= -\frac{(\tilde{a} + a)_{i+\frac{1}{2}j} + (\tilde{a} + a)_{i-\frac{1}{2}j} + (\tilde{a} + a)_{ij+\frac{1}{2}} + (\tilde{a} + a)_{ij-\frac{1}{2}}}{2\Delta\tilde{\xi}^2}
 \end{aligned}$$

$$\begin{aligned}
\mathcal{E}_{ij} &= -\frac{(\tilde{a} - a)_{i+\frac{1}{2}j} + (\tilde{a} - a)_{ij-\frac{1}{2}}}{2\Delta\tilde{\xi}^2} \\
\mathcal{SW}_{ij} &= \frac{(\tilde{a} + a)_{i-\frac{1}{2}j}}{2\Delta\tilde{\xi}^2} \\
\mathcal{S}_{ij} &= \frac{(\tilde{a} - a)_{i-\frac{1}{2}j} + (\tilde{a} - a)_{ij-\frac{1}{2}}}{2\Delta\tilde{\xi}^2} \\
\mathcal{SE}_{ij} &= \frac{(\tilde{a} + a)_{ij-\frac{1}{2}}}{2\Delta\tilde{\xi}^2}
\end{aligned}$$

where we used equation (3.3.11) to replace the elements of the mobility tensor.

3.3.3 Incorporation of Neumann boundary conditions

For Neumann boundary conditions, we specify the pressure gradient on the domain boundary. This can be expressed as

$$\frac{\partial p}{\partial \mathbf{n}}(\mathbf{x}) = \mathbf{n}(\mathbf{x}) \cdot \nabla p(\mathbf{x}) \quad (3.3.17)$$

for $\mathbf{x} \in \partial\Omega$, where \mathbf{n} is the outward pointing normal to the domain boundary. In order to explain how this condition is satisfied, we first define some useful index ranges

$$\begin{aligned}
\mathcal{I} \text{ (interior):} & \quad \{(i, j) : 1 \leq i \leq N_1 - 2, 1 \leq j \leq N_2 - 2\} \\
\mathcal{B} \text{ (bottom):} & \quad \{(i, 0) : 1 \leq i \leq N_1 - 2\} \\
\mathcal{T} \text{ (top):} & \quad \{(i, N_2 - 1) : 1 \leq i \leq N_1 - 2\} \\
\mathcal{R} \text{ (right):} & \quad \{(N_1 - 1, j) : 1 \leq j \leq N_2 - 2\} \\
\mathcal{L} \text{ (left):} & \quad \{(0, j) : 1 \leq j \leq N_2 - 2\} \\
\mathcal{CO} \text{ (corners):} & \quad \{(0, 0), (N_1 - 1, 0), (N_1 - 1, N_2 - 1), (0, N_2 - 1)\}
\end{aligned}$$

that are pairwise disjoint and satisfy $\mathcal{I} \cup \mathcal{B} \cup \mathcal{T} \cup \mathcal{L} \cup \mathcal{R} \cup \mathcal{CO} = \tilde{\Omega}$, with $\tilde{\Omega}$ being our discrete representation of Ω .

There are several ways to satisfy the boundary conditions. For the nine-point stencil, one way would be to *extend* the problem by one layer of ghost cells whose value is determined by the boundary specifications. For example, we could leave the equation for cell $(0, 0)$ unmodified and extend equation (3.3.16) with

$$p_{-1,k} = p_{0,k} + \Delta x \hat{p}_{-\frac{1}{2},k} \quad (3.3.18)$$

$$p_{k,-1} = p_{k,0} + \Delta y \hat{p}_{k,-\frac{1}{2}} \quad (3.3.19)$$

$$p_{-1,-1} = p_{0,0} + \Delta\tilde{\xi} \hat{p}_{-\frac{1}{2},-\frac{1}{2}} \quad (3.3.20)$$

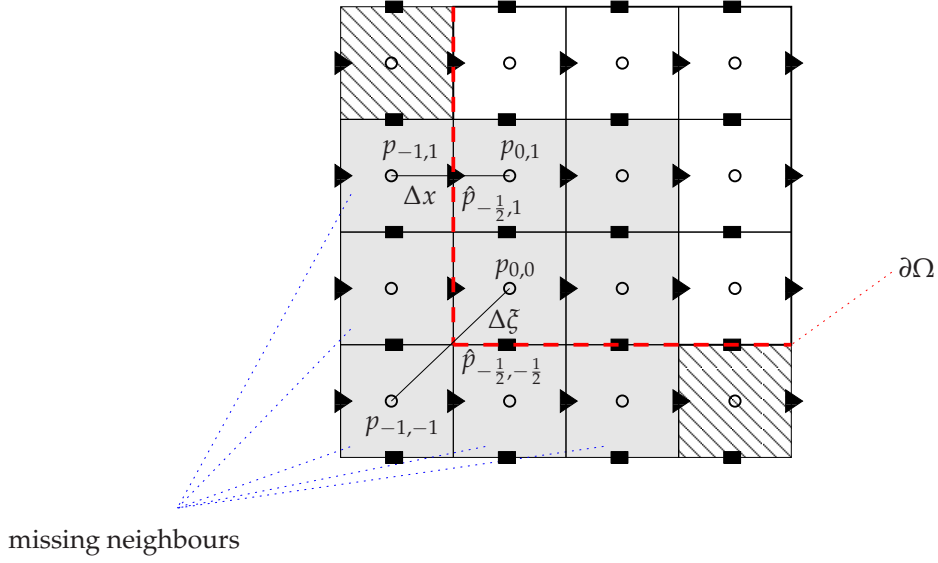


Figure 3.6: The neighbours $p_{-1,1}$, $p_{-1,0}$, $p_{-1,-1}$, $p_{0,-1}$ and $p_{1,-1}$ are missing in equation (3.3.16) for cell $(0,0)$. These values can be expressed by the derivative of p on the domain boundary $\partial\Omega$, which is denoted by \hat{p} .

for $k \in \{0,1\}$, where \hat{p} denotes the outward pointing normal component of the pressure gradient. In this case, each of the terms b_{ij} in equation (3.3.16) would be zero for $(i,j) \in \tilde{\Omega}$. In addition to the $N_1 \times N_2$ equations for the points inside the domain, another $2(N_1 + N_2) + 4$ equations would have to be solved. We can, however, eliminate these additional equations by linearly combining them with equation (3.3.16). For cell $(0,0)$, equation (3.3.16) becomes

$$\begin{aligned} \mathcal{N}\mathcal{W}_{0,0} p_{-1,1} + \mathcal{N}_{0,0} p_{0,1} + \mathcal{N}\mathcal{E}_{0,0} p_{1,1} + \mathcal{W}_{0,0} p_{-1,0} + \mathcal{C}_{0,0} p_{0,0} + \mathcal{E}_{0,0} p_{1,0} \\ + \mathcal{S}\mathcal{W}_{0,0} p_{-1,-1} + \mathcal{S}_{0,0} p_{0,-1} + \mathcal{S}\mathcal{E}_{0,0} p_{1,-1} = -q_{0,0} + b_{0,0} \end{aligned} \quad (3.3.21)$$

with $b_{0,0}$ yet to be defined. Using equations (3.3.18)–(3.3.20), we can cast this into

$$\begin{aligned} (\mathcal{C}_{0,0} + \mathcal{W}_{0,0} + \mathcal{S}_{0,0} + \mathcal{S}\mathcal{W}_{0,0}) p_{0,0} + (\mathcal{E}_{0,0} + \mathcal{S}\mathcal{E}_{0,0}) p_{1,0} \\ + (\mathcal{N}_{0,0} + \mathcal{N}\mathcal{W}_{0,0}) p_{0,1} + \mathcal{N}\mathcal{E}_{0,0} p_{1,1} = -q_{0,0} - \Delta x (\mathcal{N}\mathcal{W}_{0,0} \hat{p}_{-\frac{1}{2},1} \\ + \mathcal{N}\mathcal{E}_{0,0} \hat{p}_{-\frac{1}{2},0}) - \Delta y (\mathcal{S}_{0,0} \hat{p}_{0,-\frac{1}{2}} + \mathcal{S}\mathcal{E}_{0,0} \hat{p}_{1,-\frac{1}{2}}) - \Delta \xi \mathcal{S}\mathcal{W}_{0,0} \hat{p}_{-\frac{1}{2},-\frac{1}{2}} \end{aligned} \quad (3.3.22)$$

with

$$\begin{aligned} b_{0,0} = -\Delta x (\mathcal{N}\mathcal{W}_{0,0} \hat{p}_{-\frac{1}{2},1} + \mathcal{N}\mathcal{E}_{0,0} \hat{p}_{-\frac{1}{2},0}) \\ - \Delta y (\mathcal{S}_{0,0} \hat{p}_{0,-\frac{1}{2}} + \mathcal{S}\mathcal{E}_{0,0} \hat{p}_{1,-\frac{1}{2}}) - \Delta \xi \mathcal{S}\mathcal{W}_{0,0} \hat{p}_{-\frac{1}{2},-\frac{1}{2}}. \end{aligned} \quad (3.3.23)$$

In this case, the ghost cell values are not part of our solution vector \mathbf{p} in equation (3.3.2). We treated the most challenging case, setting up the equation for a corner cell, in the same manner as in the previous example. The equations for the remaining cells $(i, j) \in \tilde{\Omega} \setminus (\mathcal{I} \cup \{(0, 0)\})$ are treated analogously. For the cells $(i, j) \in \mathcal{B} \cup \mathcal{T} \cup \mathcal{L} \cup \mathcal{R}$, only three neighbouring cells are missing and the equations for the interior $(i, j) \in \mathcal{I}$ are given by equation (3.3.16) with $b_{ij} = 0$.

Neumann boundary conditions define the solution up to an additive constant. We found that this leads to a high condition number of $\sim 10^7$ for a 10×10 grid. The condition number of a numerical problem relates the change in the output to perturbations in the input data [4]. A high condition number means that small perturbations in the input data can have a strong effect on the solution. The situation can be improved by specifying the gradient *and* the value of p for one cell. If we do this for cell $(0, 0)$, we can regulate the value of p with a parameter κ if the condition

$$\kappa = p_{0,0} + q_{0,0} \quad (3.3.24)$$

is enforced simultaneously with equation (3.3.22). Again, we can linearly combine the equations to get

$$\begin{aligned} & (\mathcal{C}_{0,0} + \mathcal{W}_{0,0} + \mathcal{S}_{0,0} + \mathcal{SW}_{0,0} - 1) p_{0,0} + (\mathcal{E}_{0,0} + \mathcal{SE}_{0,0}) p_{1,0} \\ & + (\mathcal{N}_{0,0} + \mathcal{NW}_{0,0}) p_{0,1} + \mathcal{NE}_{0,0} p_{1,1} = -\kappa - \Delta x (\mathcal{NW}_{0,0} \hat{p}_{-\frac{1}{2},1} \\ & + \mathcal{NE}_{0,0} \hat{p}_{-\frac{1}{2},0}) - \Delta y (\mathcal{S}_{0,0} \hat{p}_{0,-\frac{1}{2}} + \mathcal{SE}_{0,0} \hat{p}_{1,-\frac{1}{2}}) - \Delta \xi \mathcal{SW}_{0,0} \hat{p}_{-\frac{1}{2},-\frac{1}{2}} \end{aligned} \quad (3.3.25)$$

and only a single equation needs to be solved. This trick has a remarkable influence on the condition number. For the same system it dropped to $\sim 10^2$, making the system of equations less sensitive to perturbations.

The final step is to weight the contributions of both stencils by ν , as illustrated in Fig. 3.3b. Since the equations are linear, we can rewrite equation (3.3.2) as

$$(\nu \mathbf{A}^0 + (1 - \nu) \mathbf{A}^{45}) \mathbf{p} = -(\nu \mathbf{q}^0 + (1 - \nu) \mathbf{q}^{45}) + (\nu \mathbf{b}^0 + (1 - \nu) \mathbf{b}^{45}) \quad (3.3.26)$$

for $\nu \in [0, 1]$ where superscripts 0 and 45 denote the contributions from the unrotated and rotated coordinate system, respectively. As for the RHS in equation (3.3.26) the evaluation is straightforward. The system of equations (3.3.26) is solved with a preconditioned, biconjugate gradient stabilized (BiCGSTAB) method, which is part of the *lsolver* package [1].

3.4 CALCULATION OF THE VELOCITY

Once we have calculated the new pressure, p^{n+1} , we use it to calculate the new velocity, \mathbf{u}^{n+1} , as illustrated in Fig. 3.1. The velocity is proportional to the pressure gradient and is defined by Darcy's Law, equation (2.1.18) under the incompressibility condition, equation (2.1.17). Our weighted formulation of the linear set of equations for the pressure also requires a weighted calculation of the velocity contributions. More specifically, if we were to calculate the velocity using the pressure gradient with respect to (x^1, x^2) -coordinates only, we would not satisfy the gauge condition, equation (2.1.17). The correct velocity is given by

$$\mathbf{u} = \nu \mathbf{u}^0 + (1 - \nu) \mathbf{u}^{45}, \quad (3.4.1)$$

taking into account both contributions. We start by transforming equation (2.1.17) to the rotated coordinate system according to

$$\mathbf{u} = -\mathbf{M} \nabla p \quad \longrightarrow \quad \tilde{\mathbf{u}} = -\tilde{\mathbf{M}} \tilde{\nabla} \tilde{p} = - \begin{pmatrix} \tilde{M}^{11} \tilde{\partial}_1 \tilde{p} + \tilde{M}^{12} \tilde{\partial}_2 \tilde{p} \\ \tilde{M}^{21} \tilde{\partial}_1 \tilde{p} + \tilde{M}^{22} \tilde{\partial}_2 \tilde{p} \end{pmatrix}. \quad (3.4.2)$$

If we now apply the central difference operators, equations (3.3.13)–(3.3.14), this becomes

$$\tilde{u}_{ij} = -\Delta \xi^{-1} \left[\tilde{M}_{ij}^{11} (\tilde{p}_{i+\frac{1}{2}j} - \tilde{p}_{i-\frac{1}{2}j}) + \tilde{M}_{ij}^{12} (\tilde{p}_{ij+\frac{1}{2}} - \tilde{p}_{ij-\frac{1}{2}}) \right] \quad (3.4.3)$$

$$\tilde{v}_{ij} = -\Delta \xi^{-1} \left[\tilde{M}_{ij}^{21} (\tilde{p}_{i+\frac{1}{2}j} - \tilde{p}_{i-\frac{1}{2}j}) + \tilde{M}_{ij}^{22} (\tilde{p}_{ij+\frac{1}{2}} - \tilde{p}_{ij-\frac{1}{2}}) \right] \quad (3.4.4)$$

carrying an $\mathcal{O}(\Delta \xi^2)$ truncation error term. The calculation of \mathbf{u}_{ij}^{45} is carried out in three steps:

1. Use equations (3.4.3)–(3.4.4) to calculate the velocities $\tilde{\mathbf{u}}_{i-\frac{1}{2}j}$, $\tilde{\mathbf{u}}_{ij-\frac{1}{2}}$ and $\tilde{\mathbf{u}}_{ij+\frac{1}{2}}$ at the cell vertices (see Fig. 3.7).
2. Transform the velocities back to (x^1, x^2) -coordinates using \mathbf{R} as defined in equation (3.3.5):

$$\mathbf{u}_{i-\frac{1}{2}j-\frac{1}{2}} = \mathbf{R} \tilde{\mathbf{u}}_{i-\frac{1}{2}j}$$

$$\mathbf{u}_{i-\frac{1}{2}j+\frac{1}{2}} = \mathbf{R} \tilde{\mathbf{u}}_{ij+\frac{1}{2}}.$$

3. Take the average to get the components at the correct positions:

$$u_{i-\frac{1}{2}j} = \frac{1}{2} \left(u_{i-\frac{1}{2}j-\frac{1}{2}} + u_{i-\frac{1}{2}j+\frac{1}{2}} \right)$$

$$v_{ij-\frac{1}{2}} = \frac{1}{2} \left(v_{i-\frac{1}{2}j-\frac{1}{2}} + v_{i+\frac{1}{2}j-\frac{1}{2}} \right).$$

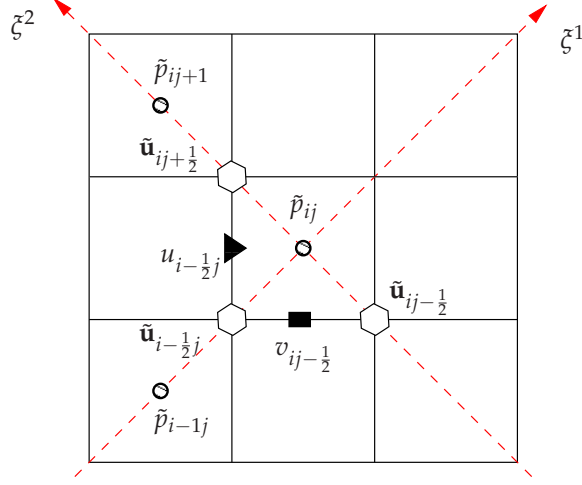


Figure 3.7: Calculation of the velocity contribution from the rotated coordinate system.

In summary, this procedure approximates the velocity

$$\mathbf{u} = -\nu \begin{pmatrix} M^{11}\partial_1 p + M^{12}\partial_2 p \\ M^{21}\partial_1 p + M^{22}\partial_2 p \end{pmatrix} - (1 - \nu) \mathbf{R} \begin{pmatrix} \tilde{M}^{11}\tilde{\partial}_1 \tilde{p} + \tilde{M}^{12}\tilde{\partial}_2 \tilde{p} \\ \tilde{M}^{21}\tilde{\partial}_1 \tilde{p} + \tilde{M}^{22}\tilde{\partial}_2 \tilde{p} \end{pmatrix} \quad (3.4.5)$$

to second-order in space, since the error terms have the same order in both coordinate systems. In the rotated coordinate system, we use one-dimensional linear interpolation to get the velocity at the desired positions (step 3 above). Given two velocities, $\mathbf{u}(\mathbf{x}_L)$ and $\mathbf{u}(\mathbf{x}_R)$, we can calculate the mean value in the centre, \mathbf{x}_C , of the line connecting the points \mathbf{x}_L and \mathbf{x}_R . Using Taylor expansion, we obtain

$$\begin{aligned} \mathbf{u}(\mathbf{x}_L) &= \mathbf{u}(\mathbf{x}_C) - (\nabla \mathbf{u} \Delta \mathbf{x})(\mathbf{x}_C) + \mathcal{O} \left(\sum_{i,j=1}^2 \Delta x^i \Delta x^j \right) \\ \mathbf{u}(\mathbf{x}_R) &= \mathbf{u}(\mathbf{x}_C) + (\nabla \mathbf{u} \Delta \mathbf{x})(\mathbf{x}_C) + \mathcal{O} \left(\sum_{i,j=1}^2 \Delta x^i \Delta x^j \right), \end{aligned}$$

where $\Delta \mathbf{x} = \mathbf{x}_C - \mathbf{x}_L = \mathbf{x}_R - \mathbf{x}_C$ and $\nabla \mathbf{u}$ denotes the Jacobian of the velocity. By adding these equations, it is easy to see that the interpolation step does not reduce the order of the overall scheme.

3.5 LEVEL SET SOLVER

In this section we first describe what level set methods are, and subsequently show how the level set function can be utilized to track the motion of an interface under an external velocity field.

3.5.1 Level set methods

Following Osher and Fedkiw [25], the basic idea of the level set method is to represent and analyze the motion of an interface, Γ , bounding a region, Ω . The interface is defined implicitly by the zero contour of a function $\Phi(\mathbf{x}, t)$, the *level set function*, which is allowed to change with time. Therefore, we can define $\Gamma(t) := \partial\Omega = \{\mathbf{x} | \Phi(\mathbf{x}, t) = 0\}$. Although this approach can be generalized to any number of spatial dimensions, we focus on the two-dimensional case. The sign of $\Phi(\mathbf{x}, t)$ determines whether the point \mathbf{x} lies inside or outside the region Ω . Fig. 3.8 illustrates this with a simple example, in which Φ is negative inside the circle, vanishes at the boundary, and is positive outside. We will follow this convention throughout the report.

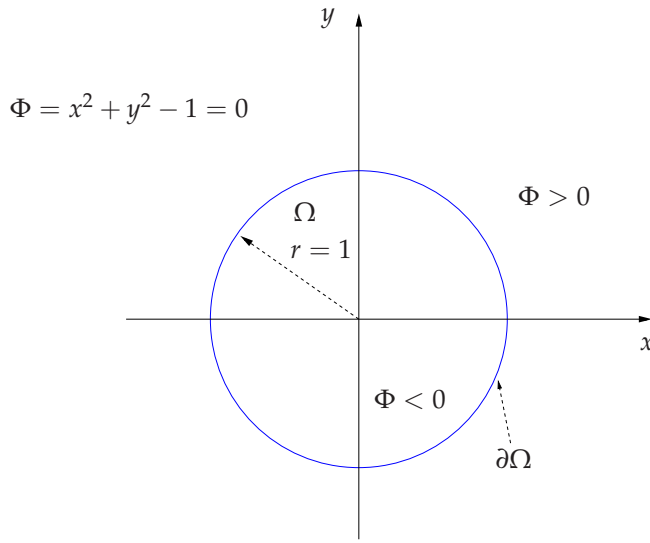


Figure 3.8: Implicit representation of a circle with radius 1 (redrawn from [25]). The level set function Φ is negative inside the domain Ω , vanishes on the boundary, and is positive outside.

The level set function is not unique; for the example in Fig. 3.8, we would not lose any information if we multiplied $\Phi(\mathbf{x}, t)$ by an arbitrary positive constant. In addition to defining the interior, the boundary and the exterior of a region, the level set function can be chosen to allow the closest distance from a point to the interface to be determined. The level set function corresponding to this choice satisfies the condition $|\nabla\Phi(\mathbf{x}, t)| = 1$ and is known as a *signed distance function*. Here, we are interested only in the position of the interface and make no further use of this property. Consequently, we do not need the reinitialization as suggested in [14].

We now know how to represent our domain, but do not yet know how the interface

changes under the influence of an external velocity field. Let $\mathbf{u}(\mathbf{x}, t) = (u(\mathbf{x}, t), v(\mathbf{x}, t))^T$ denote the external, time-dependent velocity field. Each point must then satisfy the system of ODEs

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}(\mathbf{x}, t). \quad (3.5.1)$$

The equation describing the evolution of the level set function in an external velocity field $\mathbf{u}(\mathbf{x}, t)$ is a linear advection equation,

$$\frac{\partial \Phi}{\partial t} + \mathbf{u}(\mathbf{x}, t) \nabla \Phi = 0. \quad (3.5.2)$$

A commonly used approach is the so-called *method of lines* (MOL), where the discretization is carried out in two steps. We first discretize the spatial derivative, but retain a continuous time component:

$$\frac{\partial \Phi_{ij}}{\partial t} = \mathcal{L}(\Phi(t)_{kl}, t)_{ij}. \quad (3.5.3)$$

Subsequently, an ODE solver is employed to discretize the equation in time. For a good approximation, it is important to take into account the direction from which information arrives and choose the derivative operator \mathcal{L} accordingly.

3.5.2 Level set scheme

In order to stabilize the concentration front (see §4.4), we try to track its time evolution accurately. Thus, we employ the fifth-order, finite difference *weighted essentially non-oscillatory* (WENO) scheme of Jiang and Shu [15]. In their paper, they construct the fifth-order derivative interpolation by combining three third-order interpolations with appropriate weights. They choose the weights such that the total variation of the approximation with respect to their measure of smoothness is minimized [15].

Discretizing in time, we can express \mathcal{L} as

$$\mathcal{L}(\Phi_{kl}^n, \mathbf{u}_{kl}^n)_{ij} = -u_{ij}^n \mathcal{X}_{ij}^{\text{WENO5}}(\Phi_{kl}^n, \Delta x, u_{ij}^n) - v_{ij}^n \mathcal{Y}_{ij}^{\text{WENO5}}(\Phi_{kl}^n, \Delta y, v_{ij}^n) \quad (3.5.4)$$

where $\mathcal{X}_{ij}^{\text{WENO5}}$ and $\mathcal{Y}_{ij}^{\text{WENO5}}$ approximate the spatial derivatives to fifth-order. It is sufficient to describe the reconstruction for only a single direction because the other is carried out analogously.

$$\mathcal{X}_{ij}^{\text{WENO5}}(\Phi_{kl}^n, \Delta x, u_{ij}^n) = \begin{cases} \text{WENO5}(\frac{\Phi_{i-2j}-\Phi_{i-3j}}{\Delta x}, \dots, \frac{\Phi_{i+2j}-\Phi_{i+1j}}{\Delta x}) & \text{if } u_{ij}^n > 0 \\ \text{WENO5}(\frac{\Phi_{i+3j}-\Phi_{i+2j}}{\Delta x}, \dots, \frac{\Phi_{i-1j}-\Phi_{i-2j}}{\Delta x}) & \text{if } u_{ij}^n < 0 \end{cases} \quad (3.5.5)$$

If we rename the five arguments of the WENO5 function to $\Psi_{i-2}, \Psi_{i-1}, \Psi_i, \Psi_{i+1}, \Psi_{i+2}$, the reconstruction is given by

$$\begin{aligned}\Psi^0 &= \frac{1}{3}\Psi_{i-2} - \frac{7}{6}\Psi_{i-1} + \frac{11}{6}\Psi_i, \\ \Psi^1 &= -\frac{1}{6}\Psi_{i-1} + \frac{5}{6}\Psi_i + \frac{1}{3}\Psi_{i+1}, \\ \Psi^2 &= \frac{1}{3}\Psi_i + \frac{5}{6}\Psi_{i+1} - \frac{1}{6}\Psi_{i+2}\end{aligned}$$

and the final state is a linear combination of the form

$$\text{WENO5}(\Psi_{i-2}, \Psi_{i-1}, \Psi_i, \Psi_{i+1}, \Psi_{i+2}) = \omega_0\Psi^0 + \omega_1\Psi^1 + \omega_2\Psi^2. \quad (3.5.6)$$

The weights ω_i , based on the smoothness of the solution, are

$$\begin{aligned}\omega_i &= \frac{\alpha_i}{\sum_{i=1}^3 \alpha_i}, \\ \alpha_0 &= (\epsilon + \beta_0)^{-2}, \\ \alpha_1 &= 6(\epsilon + \beta_1)^{-2}, \\ \alpha_2 &= 3(\epsilon + \beta_2)^{-2}, \\ \beta_0 &= \frac{13}{12}(\Psi_{i-2} - 2\Psi_{i-1} + \Psi_i)^2 + \frac{1}{4}(\Psi_{i-2} - 4\Psi_{i-1} + 3\Psi_i)^2, \\ \beta_1 &= \frac{13}{12}(\Psi_{i-1} - 2\Psi_i + \Psi_{i+1})^2 + \frac{1}{4}(\Psi_{i+1} - \Psi_{i-1})^2, \\ \beta_2 &= \frac{13}{12}(\Psi_i - 2\Psi_{i+1} + \Psi_{i+2})^2 + \frac{1}{4}(3\Psi_i - 4\Psi_{i+1} + \Psi_{i+2})^2.\end{aligned}$$

In practice, ϵ is set to 10^{-10} to avoid division by zero. The choice of time integration scheme is limited by the velocity, which is only available at time t^n . A simple choice is forward Euler time integration:

$$\Phi_{ij}^{n+1} = \Phi_{ij}^n + \Delta t \mathcal{L}(\Phi_{kl}^n, \mathbf{u}_{kl}^n)_{ij}, \quad (3.5.7)$$

where the velocity is assumed to be constant for each integration step. This method is only first-order accurate in time. In fact, this fifth-order WENO scheme is proven to be unstable in combination with forward Euler time integration [35]. However, these instabilities have not been seen in the current work.

3.6 HYPERBOLIC SOLVER

After calculating the velocity from the pressure field, all of the initial data for solving the advection-diffusion equation (2.1.16) in Fig. 3.1 are available. Using *operator splitting*

allows for solving this equation with methods suitable for the individual components. Many sophisticated methods are available for solving homogeneous conservation laws, parabolic equations and ODEs. This approach might be preferable to deriving an unsplit method for complex problems [22]. Therefore, we split equation (2.1.16) into three separate problems with different input data.

$$\frac{\partial}{\partial t}c + \nabla \cdot (c\mathbf{u}) - \nabla \cdot \mathbf{D}\nabla c = \hat{c}q \rightarrow \begin{cases} \frac{\partial}{\partial t}c + \nabla \cdot (c\mathbf{u}) = 0 & \text{HYPERBOLIC} \\ \frac{\partial}{\partial t}c = \hat{c}q & \text{ODE} \\ \frac{\partial}{\partial t}c = \nabla \cdot \mathbf{D}\nabla c & \text{PARABOLIC} \end{cases} \quad (3.6.1)$$

In equation (3.6.1), and throughout the rest of this report, we only consider porosity equal to unity. However, the scheme would not change for different levels of porosity.

3.6.1 Splitting scheme

We will illustrate the idea of operator splitting, or the *fractional-step method*, following LeVeque [22]. Let us consider a linear PDE in one spatial dimension, of the form

$$q_t = (\mathcal{M} + \mathcal{N})q, \quad (3.6.2)$$

where \mathcal{M} and \mathcal{N} are, e.g., differential operators that do not explicitly depend on t . Then we can write the exact solution formally as

$$q(x, \Delta t) = e^{\Delta t(\mathcal{M} + \mathcal{N})}q(x, 0). \quad (3.6.3)$$

Employing the method of fractional-steps, we can split equation (3.6.3) and solve

$$q^*(x, \Delta t) = e^{\Delta t\mathcal{M}}q(x, 0) \quad (3.6.4)$$

$$q^{**}(x, \Delta t) = e^{\Delta t\mathcal{N}}q^*(x, \Delta t) = e^{\Delta t\mathcal{N}}e^{\Delta t\mathcal{M}}q(x, 0) \quad (3.6.5)$$

instead. From this we can see that q^{**} and q differ if the operators \mathcal{M} and \mathcal{N} do not commute. In general, this splitting scheme is only first-order accurate in time. However, given that both operators are second-order, modifying it slightly to

$$q^{**}(x, \Delta t) = e^{\frac{\Delta t}{2}\mathcal{N}}e^{\Delta t\mathcal{M}}e^{\frac{\Delta t}{2}\mathcal{N}}q(x, 0) \quad (3.6.6)$$

leads to a second-order scheme, called *Strang's splitting*. Following [32], another second-order scheme is given by:

$$q^{**}(x, \Delta t) = \frac{1}{2} \left(e^{\Delta t\mathcal{M}}e^{\Delta t\mathcal{N}} + e^{\Delta t\mathcal{N}}e^{\Delta t\mathcal{M}} \right) q(x, 0). \quad (3.6.7)$$

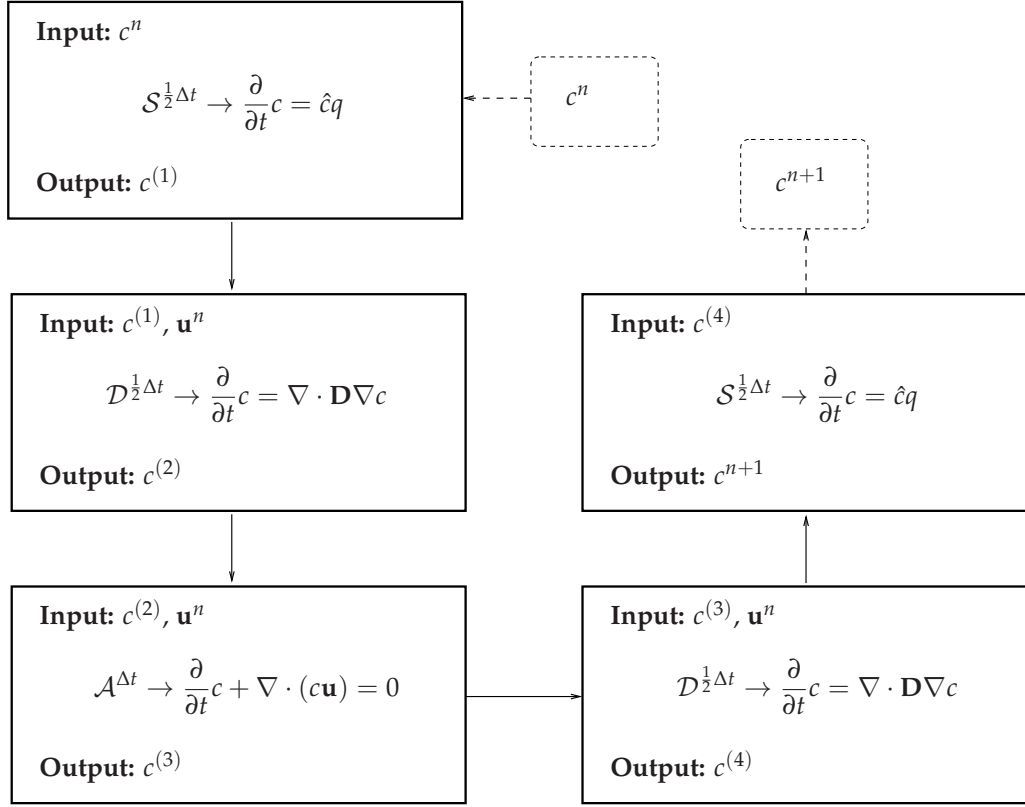


Figure 3.9: Splitting scheme: $\mathcal{S}^{\Delta t}$ solves the ODE in (3.6.1), $\mathcal{D}^{\Delta t}$ the parabolic part and $\mathcal{A}^{\Delta t}$ the hyperbolic part.

We refer to this scheme as *symmetric splitting*. Employing a second-order splitting scheme will only lead to a second-order method for the original equation if all of the operators involved are second-order accurate [22]. Furthermore, the second-order splitting schemes, equations (3.6.6) and (3.6.7) are more expensive computationally [32].

At the moment, the operator \mathcal{S} (see Fig. 3.9) is second-order accurate in time. The CTU scheme (§3.6.2.5) and the Upwind scheme (§3.6.2.1) are first-order accurate in time, while the MUSCL scheme (§3.6.2.2) and the WAF scheme (§3.6.2.4) are second-order accurate. The first-order splitting given by equation (3.6.5) is employed for the first-order schemes (see Fig. 3.10) and Strang's splitting given by equation (3.6.6) is employed for the second-order schemes (see Fig. 3.9). The same splitting is applied for the operator \mathcal{D} , but since \mathcal{D} is only first-order accurate, all of the combinations reduce to first-order accuracy in time. Nevertheless, employing second-order splitting for the source term still yields an improvement.

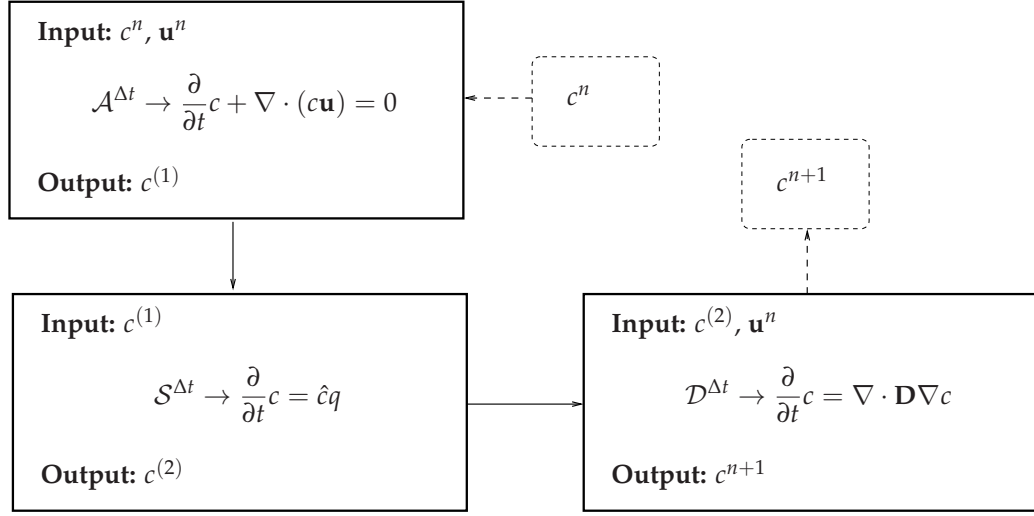


Figure 3.10: Splitting scheme: $\mathcal{S}^{\Delta t}$ solves the ODE in (3.6.1), $\mathcal{D}^{\Delta t}$ the parabolic part and $\mathcal{A}^{\Delta t}$ the hyperbolic part.

If we define operators that solve the ODE in equation (3.6.1) ($\mathcal{S}^{\Delta t}$), the parabolic part ($\mathcal{D}^{\Delta t}$) and the hyperbolic part ($\mathcal{A}^{\Delta t}$), our second-order splitting scheme can be written as

$$c^{n+1} = \mathcal{H}^{\Delta t} c^n \quad \text{with} \quad (3.6.8)$$

$$\mathcal{H}^{\Delta t} = \mathcal{S}^{\frac{1}{2}\Delta t} \mathcal{D}^{\frac{1}{2}\Delta t} \mathcal{A}^{\Delta t} \mathcal{D}^{\frac{1}{2}\Delta t} \mathcal{S}^{\frac{1}{2}\Delta t}, \quad (3.6.9)$$

and the first-order splitting as

$$c^{n+1} = \mathcal{H}^{\Delta t} c^n \quad \text{with} \quad (3.6.10)$$

$$\mathcal{H}^{\Delta t} = \mathcal{A}^{\Delta t} \mathcal{D}^{\Delta t} \mathcal{S}^{\Delta t}. \quad (3.6.11)$$

The individual components are explained in detail in the following sections.

3.6.2 Advection equation

The advection equation in equation (3.6.1) can be cast into a conservation law of the form

$$Q_t + f(Q)_x + g(Q)_y = 0 \quad (3.6.12)$$

with $Q = c$, $f(Q) = uQ$ and $g(Q) = vQ$. In this report, we solve this equation with various finite volume schemes. In the finite volume approach, the states representing cell averages are updated according to the conservative update formula either in an unsplit

or split manner [32]:

$$\text{UNSPLIT} \quad \left\{ \begin{array}{l} Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+\frac{1}{2}j}^n - \mathcal{F}_{i-\frac{1}{2}j}^n) - \frac{\Delta t}{\Delta y} (\mathcal{G}_{ij+\frac{1}{2}}^n - \mathcal{G}_{ij-\frac{1}{2}}^n) \end{array} \right. \quad (3.6.13)$$

$$\text{SPLIT} \quad \left\{ \begin{array}{l} Q_{ij}^* = Q_{ij}^n - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+\frac{1}{2}j}^n - \mathcal{F}_{i-\frac{1}{2}j}^n) \\ Q_{ij}^{n+1} = Q_{ij}^* - \frac{\Delta t}{\Delta y} (\mathcal{G}_{ij+\frac{1}{2}}^* - \mathcal{G}_{ij-\frac{1}{2}}^*). \end{array} \right. \quad (3.6.14)$$

In the latter case, the order is not necessarily the same as illustrated here. Any of the splitting schemes introduced in §3.6.1 can be employed. However, not all of them are symmetric. In order to preserve the symmetry of the problem, we therefore employ symmetric splitting for the split schemes. Each of the following schemes uses the conservative update formula and only differs in the way the fluxes are calculated. For the split schemes, we only consider the one dimensional case, since the two dimensional case follows analogously.

3.6.2.1 GODUNOV'S FIRST ORDER UPWIND SCHEME (SPLIT)

Although an unsplit version of this scheme exists [32], we are interested in the split version because this will be utilized by the more advanced *weighted average flux* (WAF) scheme. Thus, we split the two-dimensional problem into two one-dimensional problems similar to equation (3.6.14) and consider only the one dimensional-case

$$Q_t + f(Q)_x = 0. \quad (3.6.15)$$

The intercell fluxes are computed using the solutions of local Riemann problems: equation (3.6.12) must be solved at each intercell boundary for a piecewise constant distribution at the time level n in order to evolve the solution in time [32]. Let $a_{i-\frac{1}{2}}$ be the velocity at the interface. The exact solution of the local Riemann problem $\text{RP}(Q_{i-1}^n, Q_i^n)$ is then given as [32]

$$Q_{i-\frac{1}{2}}(x/t) = \begin{cases} Q_{i-1}^n & \text{if } x/t < a_{i-\frac{1}{2}}, \\ Q_i^n & \text{if } x/t > a_{i-\frac{1}{2}}. \end{cases} \quad (3.6.16)$$

With the solutions of the two Riemann problems $\text{RP}(Q_{i-1}^n, Q_i^n)$ and $\text{RP}(Q_i^n, Q_{i+1}^n)$, Godunov defined the updated solution, Q_i^{n+1} , as the integral average

$$Q_i^{n+1} = \frac{1}{\Delta x} \left(\int_0^{\frac{1}{2}\Delta x} Q_{i-\frac{1}{2}}(x/\Delta t) dx + \int_{-\frac{1}{2}\Delta x}^0 Q_{i+\frac{1}{2}}(x/\Delta t) dx \right). \quad (3.6.17)$$

With this notation, Δt is the local time in the Riemann problems' own frame of reference. The integrals can be evaluated by using equation (3.6.16), and equation (3.6.17) can be

cast into the form of equation (3.6.14) by choosing the intercell flux

$$\mathcal{F}_{i-\frac{1}{2}}^G := f(Q_{i-\frac{1}{2}}(0)) = \begin{cases} a_{i-\frac{1}{2}} Q_{i-1}^n & \text{if } a_{i-\frac{1}{2}} > 0, \\ a_{i-\frac{1}{2}} Q_i^n & \text{otherwise.} \end{cases} \quad (3.6.18)$$

This scheme is stable if and only if the CFL number

$$c = \frac{a \Delta t}{\Delta x} \quad (3.6.19)$$

satisfies $|c| \leq 1$, where a is the maximum speed. Furthermore, it is first-order accurate in both space and time [32].

3.6.2.2 MUSCL-HANCOCK SCHEME (SPLIT)

In this section we explain the *modified upwind scheme for conservation laws* (MUSCL). Following [32], the motivation for this approach is to reach higher order of accuracy through reconstruction of the data. Before, Q_i^n was defined as the integral average over the control volume. This lead to a piecewise constant distribution of the data to be advanced in time. One possible high-order extension is achieved by linearly interpolating the data with a function

$$Q_i(x) := Q_i^n + \frac{(x - x_i)}{\Delta x} \Delta_i \quad (3.6.20)$$

where $x \in [0, \Delta x]$ and x_i denotes the centre of the cell. One possible choice is

$$\Delta_i = \frac{1}{2}(1 + \omega)\Delta Q_{i-\frac{1}{2}} + \frac{1}{2}(1 - \omega)\Delta Q_{i+\frac{1}{2}}, \quad (3.6.21)$$

where

$$\Delta Q_{i-\frac{1}{2}} := Q_i^n - Q_{i-1}^n, \quad \Delta Q_{i+\frac{1}{2}} := Q_{i+1}^n - Q_i^n \quad (3.6.22)$$

and $\omega \in [-1, 1]$ is a free parameter. In these local coordinates, $x = \frac{1}{2}\Delta x$ corresponds to the centre, x_i , of the cell and thus $Q_i(x_i) = Q_i^n$. Using this linear function, we can extrapolate the values to the boundaries to get the two states

$$Q_i^L := Q_i(0) = Q_i^n - \frac{1}{2}\Delta_i, \quad Q_i^R := Q_i(\Delta x) = Q_i^n + \frac{1}{2}\Delta_i. \quad (3.6.23)$$

In the vicinity of discontinuities it proves advantageous to adapt the slope so obtained. This can be done by introducing a slope-limiter function ξ_i , in order to satisfy the *total variation diminishing* (TVD) property [32], so that

$$\bar{\Delta}_i = \xi_i \Delta_i. \quad (3.6.24)$$

A scheme which is TVD will not introduce new local extrema and hence no spurious oscillations, since this would increase the total variation [32]. The limiter function takes an argument r defined by

$$r = \frac{Q_i^n - Q_{i-1}^n}{Q_{i+1}^n - Q_i^n} \quad (3.6.25)$$

as a measure of the flow. One possible choice is the *van Leer-type slope limiter* function

$$\zeta_{lb}(r) = \begin{cases} 0 & \text{if } r \leq 0, \\ \min\left(\frac{2r}{1+r}, \zeta_R(r)\right) & \text{otherwise,} \end{cases} \quad (3.6.26)$$

where $\zeta_R(r) := 4[(1-c)(1-\omega + (1+\omega)r)]^{-1}$ and c is the CFL number [32]. The intercell flux for the MUSCL-Hancock scheme can now be calculated in three steps. First, Q_i^L and Q_i^R are calculated with the updated slope for all cells:

$$Q_i^L = Q_i^n - \frac{1}{2}\bar{\Delta}_i, \quad Q_i^R = Q_i^n + \frac{1}{2}\bar{\Delta}_i. \quad (3.6.27)$$

Second, Q_i^L and Q_i^R are evolved by a time $\frac{1}{2}\Delta t$, according to

$$\bar{Q}_i^L := Q_i^L + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(Q_i^L) - f(Q_i^R)] \quad (3.6.28)$$

$$\bar{Q}_i^R := Q_i^R + \frac{1}{2} \frac{\Delta t}{\Delta x} [f(Q_i^L) - f(Q_i^R)]. \quad (3.6.29)$$

Finally, $\mathcal{F}_{i-\frac{1}{2}}^M$ is computed by applying the flux function f to the solution of the Riemann Problem $\text{RP}(\bar{Q}_{i-1}^R, \bar{Q}_i^L)$. This scheme is second-order accurate, and is stable provided the CFL number satisfies $|c| \leq 1$ for any $\omega \in [-1, 1]$ [32]. In this report we choose ω to be zero.

3.6.2.3 WAF SCHEME (SPLIT)

Reconstructing the initial data is one way to get a second-order method for the solution of equation (3.6.15). Another approach is to employ a better approximation of the intercell flux, by averaging the flux function over a certain control volume. Following [32], the general formula of a WAF-type flux for a cell $[x_1, x_2] \times [t_1, t_2]$ in the x - t plane is

$$\mathcal{F}_{i-\frac{1}{2}}^W = \frac{1}{t_2 - t_1} \frac{1}{x_2 - x_1} \int_{t_1}^{t_2} \int_{x_1}^{x_2} f(\tilde{Q}_{i-\frac{1}{2}}(x, t)) \, dx \, dt, \quad (3.6.30)$$

where $\tilde{Q}_{i-\frac{1}{2}}(x, t)$ is the solution of a particular initial value problem. In order to obtain the original WAF-flux, the integration limits are first chosen to be $t_1 = 0, t_2 = \Delta t, x_1 = -\frac{1}{2}\Delta x$ and $x_2 = \frac{1}{2}\Delta x$ and then $\tilde{Q}_{i-\frac{1}{2}}(x, t)$ is defined to be the solution of the Riemann problem

RP(Q_{i-1}^n, Q_i^n). The approximation of the time integral in equation (3.6.30) by the midpoint rule yields

$$\mathcal{F}_{i-\frac{1}{2}}^W = \frac{1}{\Delta x} \int_{x_1}^{x_2} f\left(\tilde{Q}_{i-\frac{1}{2}}\left(x, \frac{1}{2}\Delta t\right)\right) dx, \quad (3.6.31)$$

provided Δt is such that $|c| \leq 1$. Carrying out the spatial integration in equation (3.6.31), and using a limiter so that the scheme is TVD, yields

$$\mathcal{F}_{i-\frac{1}{2}}^W := \frac{1}{2}(1 + \text{sign}(c_{i-\frac{1}{2}})\Phi_{i-\frac{1}{2}})(a_{i-\frac{1}{2}}u_{i-1}^n) + \frac{1}{2}(1 - \text{sign}(c_{i-\frac{1}{2}})\Phi_{i-\frac{1}{2}})(a_{i-\frac{1}{2}}u_i^n) \quad (3.6.32)$$

where $\Phi_{i-\frac{1}{2}}$ is a limiter function and $c_{i-\frac{1}{2}}$ is the local CFL number calculated using the velocity $a_{i-\frac{1}{2}}$ in equation (3.6.19) [32]. A possible choice for a WAF-type limiter is the *van Leer limiter* function

$$\Phi_{i-\frac{1}{2}}(r) = 1 - (1 - |c_{i-\frac{1}{2}}|)B_{vl}(r) \quad (3.6.33)$$

with

$$B_{vl}(r) = \begin{cases} 0 & \text{if } r \leq 0, \\ 2r(1+r)^{-1} & \text{otherwise} \end{cases} \quad (3.6.34)$$

and

$$r = \begin{cases} \frac{Q_{i-1}^n - Q_{i-2}^n}{Q_i^n - Q_{i-1}^n} & \text{if } c_{i-\frac{1}{2}} > 0, \\ \frac{Q_{i+1}^n - Q_i^n}{Q_i^n - Q_{i-1}^n} & \text{otherwise.} \end{cases} \quad (3.6.35)$$

3.6.2.4 WAF SCHEME (UNSPLIT)

Instead of solving two one-dimensional problems of the form (3.6.15) and updating the states sequentially with equation (3.6.14), we now want to calculate the fluxes \mathcal{F} and \mathcal{G} for both directions and carry out the update in one step with equation (3.6.13). Following [32], we define two operators

$$L_{s, dt}^G(Z^n) := Z^n + \frac{dt}{\Delta s} \left[K_{l-\frac{1}{2}}^G - K_{l+\frac{1}{2}}^G \right] \quad (3.6.36)$$

$$L_{s, dt}^W(Z_L, Z_R) := \frac{1}{\Delta s} \int_{-\frac{1}{2}\Delta s}^{\frac{1}{2}\Delta s} f\left(Z_{LR}\left(\frac{s}{dt}\right)\right) ds. \quad (3.6.37)$$

The first operator evolves the state Z^n in time by employing the upwind flux as defined in equation (3.6.18), here denoted by K , in the direction s . The second operator gives a WAF-type flux, as defined by equation (3.6.32), by calculating the integral average of a flux f in the s -direction, normal to the interface. The argument Z_{LR} is the solution to the Riemann problem RP(Z_L, Z_R). Applying these operators sequentially yields the fluxes

$$\mathcal{F}_{i-\frac{1}{2}}^{UW} := L_{x, \frac{1}{2}\Delta t}^W \left(L_{y, \frac{1}{2}\Delta t}^G(Q_{i-1j}^n), L_{y, \frac{1}{2}\Delta t}^G(Q_{ij}^n) \right) \quad (3.6.38)$$

$$\mathcal{G}_{ij-\frac{1}{2}}^{UW} := L_{y, \frac{1}{2}\Delta t}^W \left(L_{x, \frac{1}{2}\Delta t}^G(Q_{ij-1}^n), L_{x, \frac{1}{2}\Delta t}^G(Q_{ij}^n) \right), \quad (3.6.39)$$

as required by the unsplit conservative update formula, equation (3.6.13). This method is second-order accurate in space and time. It is stable if $\max(c_x, c_y) \leq 1$, where c_s is defined by equation (3.6.19) with the corresponding velocities [32].

3.6.2.5 CTU SCHEME

The previously defined split methods do not necessarily retain all the information about the directionality of the flow. One of the schemes that does take into account diagonal transport is the *corner-transport upwind* (CTU) scheme of Colella. This scheme is first-order accurate in space and time [22]. The wave propagation method of LeVeque [22], can reproduce the CTU scheme when applied to the linear advection equation. In this section, we briefly explain the wave propagation formulation of the CTU scheme. We choose this formulation because LeVeque also suggests correction terms to make it fully second-order accurate [21], which might be of interest for further investigations. Explaining the method in detail is beyond the scope of this report and the reader is referred to [21] for a detailed discussion. We merely want to illustrate the idea by the example given in Fig. 3.11.

In this example, we assume positive velocities, $u > 0$ and $v > 0$, and first consider the interface between Q_{i-1j} and Q_{ij} (solid red line). In the upwind scheme we assume that there is only a single wave, which propagates into cell Q_{ij} . However, this cell also affects Q_{ij+1} , as illustrated by the shaded area in Fig. 3.11. More precisely, it affects the flux transverse to the interface between Q_{i-1j} and Q_{ij} , leading to a correction denoted by $\tilde{\mathcal{G}}_{ij+\frac{1}{2}}$. Likewise, the interface between Q_{ij-1} and Q_{ij} (solid blue line) has an effect on the flux $\mathcal{F}_{i+\frac{1}{2}j}$. Before presenting the intercell fluxes, we consider a scalar quantity λ , and define

$$\lambda^+ := \max(\lambda, 0) \quad , \quad \lambda^- := \min(\lambda, 0). \quad (3.6.40)$$

At the beginning of each time step, we first initialize the flux corrections $\tilde{\mathcal{F}}_{i+\frac{1}{2}j}$ and $\tilde{\mathcal{G}}_{ij+\frac{1}{2}}$ to zero everywhere. Subsequently, we add all the contributions from the neighbouring cells, which can be written as

$$\begin{aligned} \tilde{\mathcal{F}}_{i-\frac{1}{2}j-1} &:= \tilde{\mathcal{F}}_{i-\frac{1}{2}j-1} - \frac{\Delta t}{2\Delta y} u_{i-\frac{1}{2}j-1}^- v_{ij-\frac{1}{2}}^- (Q_{ij} - Q_{ij-1}) \\ \tilde{\mathcal{F}}_{i+\frac{1}{2}j-1} &:= \tilde{\mathcal{F}}_{i+\frac{1}{2}j-1} - \frac{\Delta t}{2\Delta y} u_{i+\frac{1}{2}j-1}^+ v_{ij-\frac{1}{2}}^- (Q_{ij} - Q_{ij-1}) \\ \tilde{\mathcal{F}}_{i-\frac{1}{2}j} &:= \tilde{\mathcal{F}}_{i-\frac{1}{2}j} - \frac{\Delta t}{2\Delta y} u_{i-\frac{1}{2}j}^- v_{ij-\frac{1}{2}}^+ (Q_{ij} - Q_{ij-1}) \\ \tilde{\mathcal{F}}_{i+\frac{1}{2}j} &:= \tilde{\mathcal{F}}_{i+\frac{1}{2}j} - \frac{\Delta t}{2\Delta y} u_{i+\frac{1}{2}j}^+ v_{ij-\frac{1}{2}}^+ (Q_{ij} - Q_{ij-1}) \end{aligned} \quad (3.6.41)$$

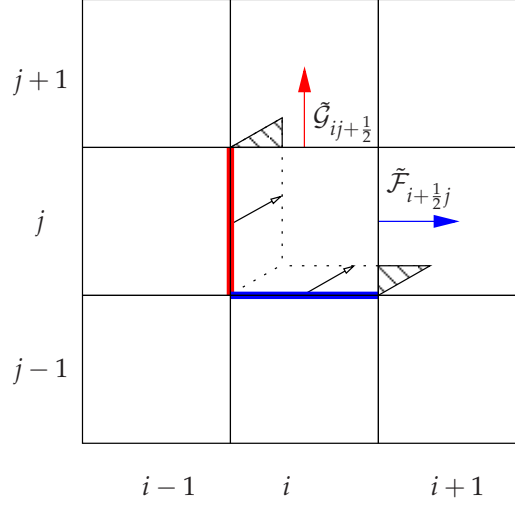


Figure 3.11: Correction of the fluxes $\mathcal{F}_{i+\frac{1}{2}j}$ and $\mathcal{G}_{ij+\frac{1}{2}}$ (redrawn from [22] and modified).

and

$$\tilde{\mathcal{G}}_{i-1j-\frac{1}{2}} := \tilde{\mathcal{G}}_{i-1j-\frac{1}{2}} - \frac{\Delta t}{2\Delta x} v_{i-1j-\frac{1}{2}}^- u_{i-\frac{1}{2}j}^- (Q_{ij} - Q_{i-1j}) \quad (3.6.42)$$

$$\tilde{\mathcal{G}}_{i-1j+\frac{1}{2}} := \tilde{\mathcal{G}}_{i-1j+\frac{1}{2}} - \frac{\Delta t}{2\Delta x} v_{i-1j+\frac{1}{2}}^+ u_{i-\frac{1}{2}j}^- (Q_{ij} - Q_{i-1j})$$

$$\tilde{\mathcal{G}}_{ij-\frac{1}{2}} := \tilde{\mathcal{G}}_{ij-\frac{1}{2}} - \frac{\Delta t}{2\Delta x} v_{ij-\frac{1}{2}}^- u_{i-\frac{1}{2}j}^+ (Q_{ij} - Q_{i-1j})$$

$$\tilde{\mathcal{G}}_{ij+\frac{1}{2}} := \tilde{\mathcal{G}}_{ij+\frac{1}{2}} - \frac{\Delta t}{2\Delta x} v_{ij+\frac{1}{2}}^+ u_{i-\frac{1}{2}j}^+ (Q_{ij} - Q_{i-1j})$$

where $:=$ denotes an update in this case [22]. Finally, the fluxes for the wave propagation method are defined by

$$\mathcal{F}_{i-\frac{1}{2}j}^{WP} := \mathcal{F}_{i-\frac{1}{2}j}^G + \tilde{\mathcal{F}}_{i-\frac{1}{2}j} \quad (3.6.43)$$

$$\mathcal{G}_{ij-\frac{1}{2}}^{WP} := \mathcal{G}_{ij-\frac{1}{2}}^G + \tilde{\mathcal{G}}_{ij-\frac{1}{2}} \quad (3.6.44)$$

and the scheme is stable for $\max(c_x, c_y) \leq 1$ [22].

3.6.3 Source term

For solving the ODE in equation (3.6.1), we use the trapezoidal rule. Consider a system of ODEs

$$\frac{\partial \mathbf{y}}{\partial t} = \mathbf{f}(t, \mathbf{y}), \quad t \geq 0, \quad (3.6.45)$$

where $\mathbf{y} \in \mathbb{R}^N$ and the function $\mathbf{f} : \mathbb{R} \times \mathbb{R}^N \rightarrow \mathbb{R}$. The trapezoidal rule is then given by [4]

$$\mathbf{y}^{n+1} = \mathbf{y}^n + \frac{1}{2}\Delta t \left[\mathbf{f}(t^n, \mathbf{y}^n) + \mathbf{f}(t^{n+1}, \mathbf{y}^{n+1}) \right] \quad (3.6.46)$$

with initial conditions $\mathbf{y}^0 := \mathbf{y}(t = 0)$ and $t^n = n\Delta t$. This scheme is the second-order A-stable method with the smallest truncation error. A-stability implies that it is not necessary to decrease the time step for stability but only for accuracy [4]. Applying the trapezoidal rule to our system yields

$$c_{ij}^{n+1} = \begin{cases} c_{ij}^n (1 + \frac{1}{2}\Delta t q_{ij})(1 - \frac{1}{2}\Delta t q_{ij})^{-1} & \text{if } q_{ij} < 0, \\ c_{ij}^n + \Delta t q_{ij} & \text{if } q_{ij} > 0, \\ c_{ij}^n & \text{otherwise,} \end{cases} \quad (3.6.47)$$

which is in explicit form.

3.6.4 Parabolic equation

In this section we approximate an equation of the form

$$\frac{\partial}{\partial t} c = \nabla \cdot \begin{pmatrix} d^{11} & d^{12} \\ d^{21} & d^{22} \end{pmatrix} \nabla c \quad (3.6.48)$$

where the elements of the tensor satisfy $d^{kk} \geq 0$ for $k = 1, 2$ and $d^{21} = d^{12}$. Assuming square grid cells and applying the method of lines, we approximate the spatial derivatives with central differences as defined in equation (3.3.13). Subsequently using forward Euler time integration yields

$$\begin{aligned} c_{ij}^{n+1} = c_{ij}^n + \frac{\Delta t}{\Delta x^2} & \left[d_{i+\frac{1}{2}j}^{11} (c_{i+1j}^n - c_{ij}^n) + d_{i+\frac{1}{2}j}^{12} (c_{i+\frac{1}{2}j+\frac{1}{2}}^n - c_{i+\frac{1}{2}j-\frac{1}{2}}^n) \right. \\ & - d_{i-\frac{1}{2}j}^{11} (c_{ij}^n - c_{i-1j}^n) - d_{i-\frac{1}{2}j}^{12} (c_{i-\frac{1}{2}j+\frac{1}{2}}^n - c_{i-\frac{1}{2}j-\frac{1}{2}}^n) \\ & + d_{ij+\frac{1}{2}}^{21} (c_{i+\frac{1}{2}j+\frac{1}{2}}^n - c_{i-\frac{1}{2}j+\frac{1}{2}}^n) + d_{ij+\frac{1}{2}}^{22} (c_{ij+1}^n - c_{ij}^n) \\ & \left. - d_{ij-\frac{1}{2}}^{21} (c_{i+\frac{1}{2}j-\frac{1}{2}}^n - c_{i-\frac{1}{2}j-\frac{1}{2}}^n) - d_{ij-\frac{1}{2}}^{22} (c_{ij}^n - c_{ij-1}^n) \right], \end{aligned} \quad (3.6.49)$$

where

$$c_{i-\frac{1}{2}j-\frac{1}{2}} = \frac{1}{4}(c_{i-1j-1} + c_{i-1j} + c_{ij} + c_{ij-1}) \quad (3.6.50)$$

is the arithmetic mean. Shubin and Bell [29] suggest using a rotationally invariant dispersion tensor of the form [29]

$$\mathbf{D} = \frac{\alpha_l}{|\mathbf{u}|} \begin{pmatrix} u^2 & uv \\ uv & v^2 \end{pmatrix} + \frac{\alpha_t}{|\mathbf{u}|} \begin{pmatrix} v^2 & -uv \\ -uv & u^2 \end{pmatrix} \quad (3.6.51)$$

where the coefficients α_l and α_t regulate the magnitudes of longitudinal and transverse dispersion [12]. With these coefficients we can define the longitudinal and transverse Péclet numbers

$$Pe_l = \frac{L}{\alpha_l} \quad \text{and} \quad Pe_t = \frac{L}{\alpha_t}, \quad (3.6.52)$$

respectively, where L is the distance between the corners of the injection and the production well in Fig. 2.1a [12]. In our considerations, α_l and α_t are both of order $\mathcal{O}(\Delta x)$ which implies that all the elements d^{kl} are of order $\mathcal{O}(\Delta x)$.

The discretization given by equation (3.6.49) imposes a constraint on the maximum stable time step. In order to estimate this time step, we require our method to be *monotone*. We will follow [20] for the definition of a monotone method. Consider a numerical method of the form $U_j^{n+1} = \mathcal{H}(U^n; j)$. This method is called a monotone method, if the property

$$V_j^n \geq U_j^n \quad \forall j \quad \Rightarrow \quad V_j^{n+1} \geq U_j^{n+1} \quad \forall j \quad (3.6.53)$$

holds. In order to prove that a method has this property, it is sufficient to check that [20]

$$\frac{\partial}{\partial U_i^n} \mathcal{H}(U^n; j) \geq 0 \quad \forall i, j, U^n. \quad (3.6.54)$$

Motivated by this one-dimensional definition, we try to derive a time step restriction by requiring that

$$\frac{\partial}{\partial c_{ij}^n} \mathcal{H}(c^n; k, l) \geq 0 \quad \forall i, j, k, l, c^n \quad (3.6.55)$$

holds, which is the same as requiring that all the coefficients of the values c_{ij}^n in equation (3.6.49) are positive. The values d^{kl} are of order $\mathcal{O}(\Delta x)$ and additionally they get multiplied by $\frac{\Delta t}{\Delta x^2}$. Thus, we can linearize d^{kl} around the cell centre, because $\mathcal{O}(\Delta x^2)$ terms cancel out with the denominator and the product is of order $\mathcal{O}(\Delta t)$. Additionally, we can replace the averages $c_{i-\frac{1}{2}j-\frac{1}{2}}$ by the corresponding cell centred values, which yields

$$\begin{aligned} c_{ij}^{n+1} = c_{ij}^n + \frac{\Delta t}{\Delta x^2} & \left[d^{11}(c_{i+1j}^n - 2c_{ij}^n + c_{i-1j}^n) + d^{22}(c_{ij+1}^n - 2c_{ij}^n + c_{ij-1}^n) \right. \\ & \left. + \frac{1}{2}d^{12}(c_{i+1j+1}^n - c_{i+1j-1}^n - c_{i-1j+1}^n + c_{i-1j-1}^n) \right], \end{aligned} \quad (3.6.56)$$

where we employed the symmetry property of \mathbf{D} . The last term in equation (3.6.56) is again of order $\mathcal{O}(\Delta x^2)$ and therefore we can ignore it. All the coefficients are nonnegative except for the one of c_{ij} . In order for the coefficient of c_{ij} to be nonnegative, the time step has to satisfy the inequality

$$1 - \frac{2\Delta t}{\Delta x^2} (d_{ij}^{11} + d_{ij}^{22}) \geq 0 \quad \forall i, j. \quad (3.6.57)$$

Thus using the numerical dispersion tensor (3.6.51), the maximum stable time step for this scheme is given by

$$\Delta t_{max}^D = \min_{ij} \frac{\Delta x^2}{2(d_{ij}^{11} + d_{ij}^{22})} = \min_{ij} \frac{\Delta x^2}{2|\mathbf{u}_{ij}|(\alpha_l + \alpha_t)} \quad (3.6.58)$$

and depends on the magnitude of transverse and longitudinal diffusion.

3.6.5 Maximum stable time step

As outlined in the previous sections, the numerical schemes impose restrictions on the time step. More specifically, the advection equation and the parabolic equation require the time step to be of the same order as the spatial resolution, i.e. $\Delta t \propto \Delta x$. We assume that the integration of the source term does not impose a time step restriction because the employed method is A-stable. Choosing a time step that is too big might lead to oscillations and the numerical solution might become unstable. Employing a time step that is too small can lead to a lot of time steps and hence to a long computation time. Thus, we want to use an efficient time step that is slightly below the stability limit.

Recall that for two dimensions the time step restriction for the advection equation is given by

$$\Delta t_{max}^A = \eta \min_{ij} \left(\frac{\Delta x}{|u_{ij}|}, \frac{\Delta y}{|v_{ij}|} \right) \quad (3.6.59)$$

where η is the CFL number. We can then choose the time step to be

$$\Delta t_{max} = \min(\lambda \Delta t_{max}^D, \Delta t_{max}^A) \quad (3.6.60)$$

where λ is a positive parameter similar to the CFL number that should satisfy $\lambda \leq 1$. Obeying the time step restriction (3.6.60) guarantees that the time step satisfies all the restrictions imposed by the set of split equations (3.6.1).

3.6.6 Convergence tests

Since we do not have a test problem with an analytical solution for the underlying set of equations (2.1.16)–(2.1.18), we validate the hyperbolic and elliptic solvers independently from each other.

3.6.6.1 HYPERBOLIC EQUATION

The hyperbolic part is tested by fully advecting an initial profile diagonally across the domain using periodic boundary conditions. The profile is chosen such that it is smooth

with respect to the boundary conditions. This is satisfied by

$$f(x, y) = 1 + \sin(2\pi x) \sin(2\pi y) \quad (3.6.61)$$

for $(x, y) \in [0, 1] \times [0, 1]$. Each cell is initialized with the exact cell average

$$\begin{aligned} \frac{1}{\Delta x^2} \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} dx \int_{y_i - \frac{\Delta x}{2}}^{y_i + \frac{\Delta x}{2}} dy f(x, y) = 1 + \frac{1}{4\pi^2 \Delta x^2} \times \\ \left[\cos \left[2\pi \left(x_i + \frac{\Delta x}{2} \right) \right] - \cos \left[2\pi \left(x_i - \frac{\Delta x}{2} \right) \right] \right] \\ \left[\cos \left[2\pi \left(y_i + \frac{\Delta x}{2} \right) \right] - \cos \left[2\pi \left(y_i - \frac{\Delta x}{2} \right) \right] \right] \end{aligned} \quad (3.6.62)$$

where (x_i, y_i) denotes the cell centre and we assume equal grid spacing in both dimensions.

3.6.6.2 PRESSURE EQUATION

In order to test the elliptic solver, we employ the *method of manufactured solutions*. In this approach, one specifies the solution and arranges the remaining terms so that the equation is satisfied. The designed solution should satisfy no-flow boundary conditions, which means that the gradient should vanish at the boundary. We can make an Ansatz

$$\phi(x, y) = \cos(x) \cos(y) \quad (3.6.63)$$

and easily verify that

$$\partial_x \phi(0, y) = 0, \quad \partial_x \phi(2\pi, y) = 0 \quad \text{and} \quad \partial_y \phi(x, 0) = 0, \quad \partial_y \phi(x, 2\pi) = 0 \quad (3.6.64)$$

hold for a domain $[0, 2\pi] \times [0, 2\pi]$. Plugging this function into the RHS of the pressure equation (3.3.3) yields

$$-\nabla \cdot \frac{\mathbf{K}}{\mu(0)} \nabla \phi = [\sin(x) \cos(y) \partial_x a + \cos(x) \sin(y) \partial_y \tilde{a} + \quad (3.6.65)$$

$$\cos(x) \cos(y) (a + \tilde{a})] =: q(x, y) \quad (3.6.66)$$

if we take the concentration to be zero. The diagonal entries of the permeability tensor are chosen to be

$$a(x, y) = \mu(0)^{-1} [2 + \sin(2x) \sin(y)] \quad (3.6.67)$$

$$\tilde{a}(x, y) = \mu(0)^{-1} [2 + \sin(x) \sin(2y)] \quad (3.6.68)$$

and the velocity is given by

$$\mathbf{u}(x, y) = \begin{pmatrix} a(x, y) \sin(x) \cos(y) \\ \tilde{a}(x, y) \cos(x) \sin(y) \end{pmatrix}. \quad (3.6.69)$$

3.6.6.3 CONVERGENCE RESULTS

Whether a numerical solution converges at all as well as the rate of convergence is depending on the choice of the norm [22]. The L_1 -norm is a suitable choice for conservation laws [22]. Therefore, the presented convergence studies are carried out with respect to the L_1 -norm, which is given by

$$L_1(\mathbf{e}) = \Delta x \Delta y \sum_{ij} |e_{ij}| \quad (3.6.70)$$

for the discrete grid function \mathbf{e} . The results are presented in Figs. 3.12 and 3.13, respectively.

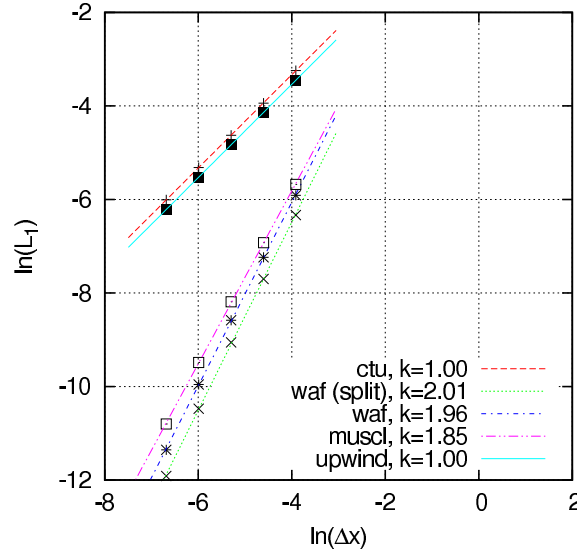


Figure 3.12: Results for the convergence test for the hyperbolic solver; k denotes the slope of the corresponding linear fit; settings: $\Delta x = \Delta y$, resolution: $N = 50 \times 2^k$ for $k = 0, \dots, 4$, parallel splitting for split schemes, $\eta = 0.9$.

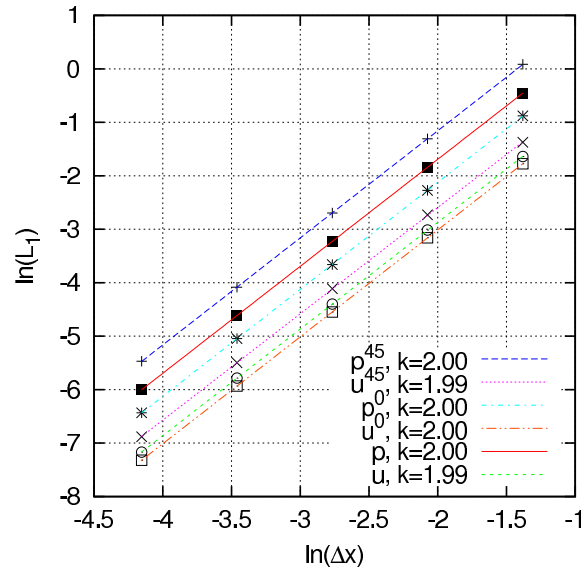


Figure 3.13: Results for the convergence test for the elliptic solver; k denotes the slope of the corresponding linear fit, u and p denote velocity and pressure for the nine-point stencil, respectively; u^0 and p^0 denote velocity and pressure for the unrotated five-point stencil; u^{45} and p^{45} denote velocity and pressure for the rotated five-point stencil; settings: $\Delta x = \Delta y$, resolution: $N = 25 \times 2^k$ for $k = 0, \dots, 4$; tolerance for the relative error (conjugate gradient method): 10^{-10} .

Numerical results

The wide range of parameters makes it necessary to focus on specific aspects rather than trying to cover all interesting phenomena of miscible displacement. As outlined in §2.1.2, the flow is destabilized by two dimensionless parameters, the Péclet number Pe and the mobility ratio M . Considerable attention has been paid to their influence on the solution in previous work. However, the resolutions being investigated were, apart from a few exceptions [8, 5, 37], lower than 100×100 cells. It is well known that numerical diffusion-dispersion only hides the effects of grid orientation up to certain resolutions [3]. Therefore, it is disappointing that even some of the more recent work does not show results for resolutions higher than 100×100 cells [17, 19].

A very detailed high-resolution study of miscible displacement in the quarter five-spot configuration was carried out by Chen and Meiburg [5], where the authors investigate the influence of Pe and M with resolutions of up to 1024×1024 cells using a high-order implicit method. The same authors compare their numerical results with actual experiments in Petitjeans *et al.* [26]. Unfortunately, in neither of the two papers do they compare the solutions obtained for the parallel and diagonal configuration. This raises the question up to what extent they were able to eliminate the GOE. Furthermore, they model the dependence of the effective viscosity on the concentration of the injected fluid differently.

In this report we only consider one fixed mobility ratio which is high enough to destabilize the flow. For the schemes discussed in the previous chapter, we try to determine the minimum amount of dispersion necessary to stabilize the flow for resolutions of up to 200×200 and 283×283 cells, respectively (see §4.3.3). These resolutions are comparable to those used in recent simulations of thermal flow in porous media [34]. For some of the schemes, we will present high-resolution results of 400×400 and 566×566 cells, respectively, and demonstrate how the GOE becomes apparent as compared to results obtained for lower resolutions. The time evolution of the error for the incompressibility condition, as well as for the no-flow boundaries and mass conservation are presented in §4.3.4.

In §4.4, we superimpose the level set function onto the concentration field and show how the initial interface develops over time. Additionally, we exploit the possibility of partially stabilizing the front using level set information. We are not aware that this approach has been considered in previous work.

From the range of hyperbolic schemes investigated in this study, only the CTU scheme and the MUSCL scheme turned out to be stable for our splitting approach. In this setting, both schemes are reliable even for a high CFL number of 0.99 in the case of $Pe_l = Pe_t = \infty$. As soon as we add dispersion, this will impose limitations on the maximum stable time step, as shown in §3.6.5. We found that the upwind scheme and the WAF scheme do not exhibit equally good stability properties and low CFL numbers were necessary for the runs without dispersion. Reducing the CFL number to 0.2 gave an improvement, but the overall error is likely due to a bug in the code or some numerical effect. Thus, we decided to focus on the CTU scheme and MUSCL scheme and use the other schemes only to demonstrate the effect.

As for the Péclet numbers, they change as we increase the resolution if we employ a numerical dispersion tensor (see equation (3.6.52)). Keeping the Péclet numbers fixed corresponds to physical dispersion. However, this is only meaningful if we take into account the various length scales on which these phenomena happen in reality.

4.1 PARAMETERS

To be consistent, we use the same input parameters for all the simulations with the exception of the choice of scheme, spatial resolution and the magnitude of dispersion. These parameters are presented in Tab. 4.1.

It is worth pointing out, that the CFL number has an effect on the solution [29]. Modified equation analysis reveals that first-order methods are diffusive and second-order methods are dispersive [20]. Here this property refers to the fact that Fourier components propagate at different speeds depending on the wave number. Thus running with a very small time step can yield more diffusive results if the numerical scheme is diffusive itself. In the context of grid orientation, employing the maximum stable time step might not be favorable because numerical diffusion could improve the results. Some authors decided to use a time step that corresponds to filling up one cell per time step [29]. We find this is unnecessarily low and employ a time step corresponding to 90% of the maximum stable time step. Using circular wells yields a lower maximum velocity as compared to point sources. Thus, larger time steps can be employed for circular wells if advection is limiting the time step. Since the high-resolution runs are computationally expensive, the

comparison is carried out for circular wells only.

4.2 DEMONSTRATION OF THE GRID ORIENTATION EFFECT

Recall that due to the symmetric configuration of the wells we expect the same results for the overlapping region of the diagonal and the parallel domain in Fig. 2.1a. In order to compare the results for both configurations, the parallel solution is projected onto the diagonal grid as illustrated in Fig. 2.2. Different resolutions are chosen for both configurations to obtain the same grid spacing. In this section we first illustrate the GOE as observed for the WAF scheme and the upwind scheme without numerical dispersion. Subsequently, we demonstrate how the effects of grid orientation change as we increase the magnitude of dispersion for the CTU scheme and the MUSCL scheme.

Let us first consider the solutions for the upwind scheme presented in Fig. 4.1a-b. The fluid is injected in the left bottom corner and produced in the right top corner. Somehow contrary to intuition, in the diagonal configuration the injected fluid starts spreading towards the left top corner and right bottom corner initially. Later on in the simulation we can observe a trend towards the production well resulting in three major fingers. In the parallel configuration the solution behaves like radial source flow early in the simulation which is consistent with the observations of Chen and Meiburg [5]. Afterwards a single finger starts forming and breakthrough is reached within the simulation time.

As for the WAF scheme, many fingers form in the diagonal configuration in Fig. 4.1c. Since the scheme is second-order accurate and TVD, it exhibits little numerical dispersion as compared to the first-order scheme [6] and hence fingers can form more easily. Furthermore, we can observe a stronger trend towards the production well in Fig. 4.1c as compared to Fig. 4.1a. In the parallel configuration (Fig. 4.1d) several fingers merge to form two major fingers that join at the production well. From Figs. 4.1a-d we can clearly see, that the GOE as exhibited by these schemes does not allow for reliably solving the underlying equations with respect to equivalent well configurations.

4.3 COMPARISON FOR CTU AND MUSCL

Having illustrated the GOE, we continue with the results for the CTU scheme and the MUSCL scheme for various amounts of numerical dispersion. In §4.3.1 the effect becomes most apparent as we do not add numerical dispersion. Subsequently, we present results for the numerical dispersion tensor employed by Shubin and Bell [29] in §4.3.2.

Parameter	Value	Units	Description
N	varying		Spatial resolution for each dimension yielding a total number of N^2 cells.
Ω	$[0, X] \times [0, X]$		Physical domain.
X	$1.0 \mid \sqrt{2}$	l	Edge length of the quarter five-spot domain (diagonal \mid parallel).
L	$\sqrt{2}$	l	Distance between a pair of neighbouring injection and production wells (diagonal \mid parallel).
I	$1.0 \mid -1.0$	$l^2 t^{-1}$	Injection rate or volumetric flow rate (injection well \mid production well).
t_{final}	0.4	t	End time of the simulation which corresponds to the injection of 40% of the total pore volume.
ϵ	10^{-8}		Relative tolerance of the elliptic solver defined by $ \mathbf{Ax} - \mathbf{b} < \epsilon \mathbf{b} $.
ϕ	1.0		Porosity of the porous medium.
ν	$\frac{2}{3}$		Weighting factor between the two stencils for the elliptic solver as defined in equation (3.3.26).
η	0.9*		CFL number as defined in equation (3.6.59).
λ	0.9		Positive parameter limiting the maximum diffusion time step as defined in equation (3.6.60).
α_l	varying	l	Represents the magnitude of longitudinal dispersion per average velocity magnitude.
α_t	varying	l	Represents the magnitude of transverse dispersion per average velocity magnitude.
$\mu_{\tilde{c}}$	41	$ml^{-1}t^{-1}$	Viscosity of the resident fluid.
μ_c	1	$ml^{-1}t^{-1}$	Viscosity of the injected fluid.
\mathbf{K}	diag(1, 1)	l^2	Permeability of the porous medium.
M	41		Mobility ratio.
Pe_t	varying		Transverse Péclet number.
Pe_l	varying		Longitudinal Péclet number.
well	circular		Well model as described in §2.3.
r_C	$0.05 \mid 0.07$	l	Cut-off radius for the circular well (diagonal \mid parallel).

Table 4.1: Input parameters for the simulations; m denotes unit of mass, l unit of length and t unit of time; *) for the upwind scheme and the WAF scheme we found it necessary to reduce the CFL number to 0.2 for stability reasons.

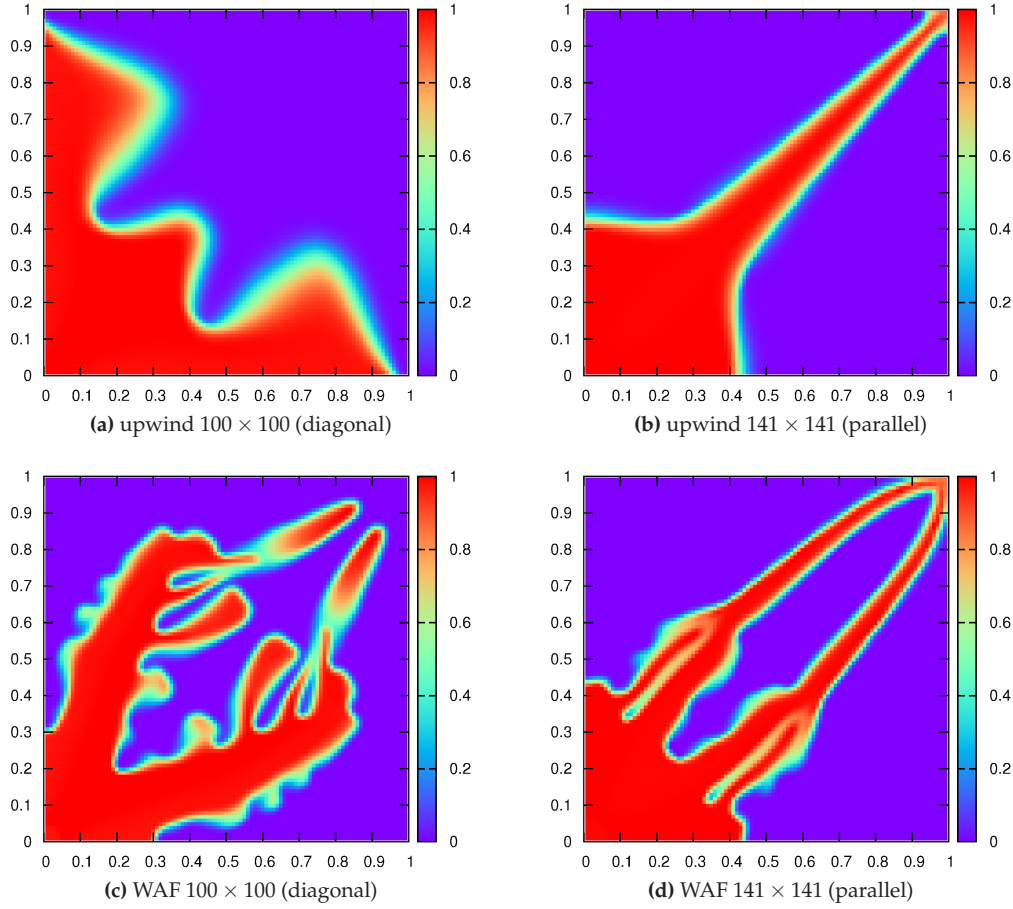


Figure 4.1: Demonstration of the GOE in the absence of dispersion ($Pe_l = Pe_t = \infty$); left column: diagonal solutions; right column: parallel solutions.

Finally, we show some improved results for our new tensor with higher magnitudes of dispersion in §4.3.3.

4.3.1 No numerical dispersion

Our first observation is that the shapes of the results change as we double the resolution (Figs. 4.2a,c and 4.2b,d, respectively). This is why we forego calculating error norms with respect to high-resolution runs. In Fig. 4.2a we can see the formation of three fingers, where the central finger is more pronounced as compared to Fig. 4.1a. The central finger splits into two when we increase the resolution (Fig. 4.2c).

Both parallel solutions exhibit three central fingers. In the lower resolution case, break-

through is achieved through the outer fingers which join at the production well. Whereas for higher resolution only the central finger reaches the production well and the side fingers are less pronounced. The difference in the results between the two configurations is considerable.

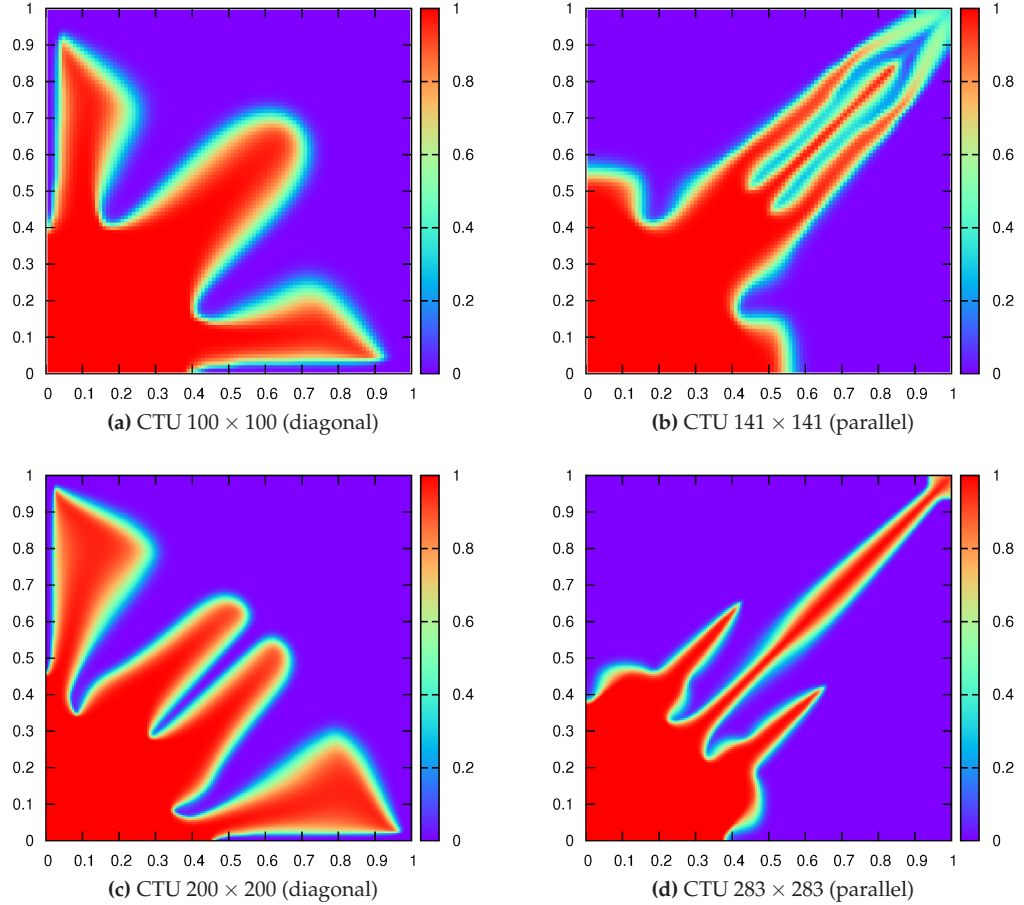


Figure 4.2: Results for the CTU scheme in the absence of dispersion ($Pe_l = Pe_t = \infty$); left column: diagonal solutions; right column: parallel solutions.

The low-resolution results for the MUSCL scheme are similar to those observed for the WAF scheme. More fingers tend to form if the resolution is increased (Figs. 4.3c,d). This time breakthrough occurs in the diagonal case as well (Fig. 4.3c). Going to higher resolutions of 400×400 and 566×566 cells, respectively, does not improve the considerable GOE exhibited by this scheme (Figs. 4.3e,f). However, we can see a slight improvement as compared to the results for the CTU scheme in that the diagonal results show a stronger trend towards the production well.

4.3.2 Numerical dispersion of Shubin and Bell

In the previous section we demonstrated the case $\mathbf{D} \equiv 0$. Now we consider a numerical dispersion tensor suggested by Shubin and Bell [29]. This tensor is based on modified equation analysis and lead to significant improvements in their study of miscible displacement [29]. We denote the tensor by \mathbf{D}^{SB} . It corresponds to equation (3.6.51) with coefficients $\alpha_l = \frac{1}{2}\Delta x$ and $\alpha_t = \frac{1}{4}\Delta x$ in the case of square grid blocks. Equation (3.6.51) then simplifies to

$$\mathbf{D}^{SB} = \frac{\Delta x}{2|\mathbf{u}|} \begin{pmatrix} u^2 + \frac{1}{2}v^2 & \frac{1}{2}uv \\ \frac{1}{2}uv & v^2 + \frac{1}{2}u^2 \end{pmatrix} \quad (4.3.1)$$

where all coefficients are of order $\mathcal{O}(\Delta x)$ in space.

Applying dispersion has a strong effect on the solution, as can be seen in Figs. 4.4 and 4.5, respectively. The formation of fingers is reduced as compared to the previous results. However for neither of the schemes the amount of dispersion is big enough to yield similar solutions for the diagonal and parallel configuration, respectively. In case of the CTU scheme, the three fingers observed for the parallel configuration (Figs. 4.2b,d) are smeared out to form one big finger (Figs. 4.4b,d). The two central fingers exhibited in the diagonal configuration (Fig. 4.2b) are not separated anymore.

The effects of dispersion become even more apparent for the MUSCL scheme. The fine fingers exhibited in the high-resolution case (Figs. 4.3c,d) are suppressed completely (Figs. 4.4c,d) as dispersion is applied. Although there is a better agreement for lower resolutions (Figs. 4.5a,b), the results exhibit a considerable discrepancy as the grid is refined.

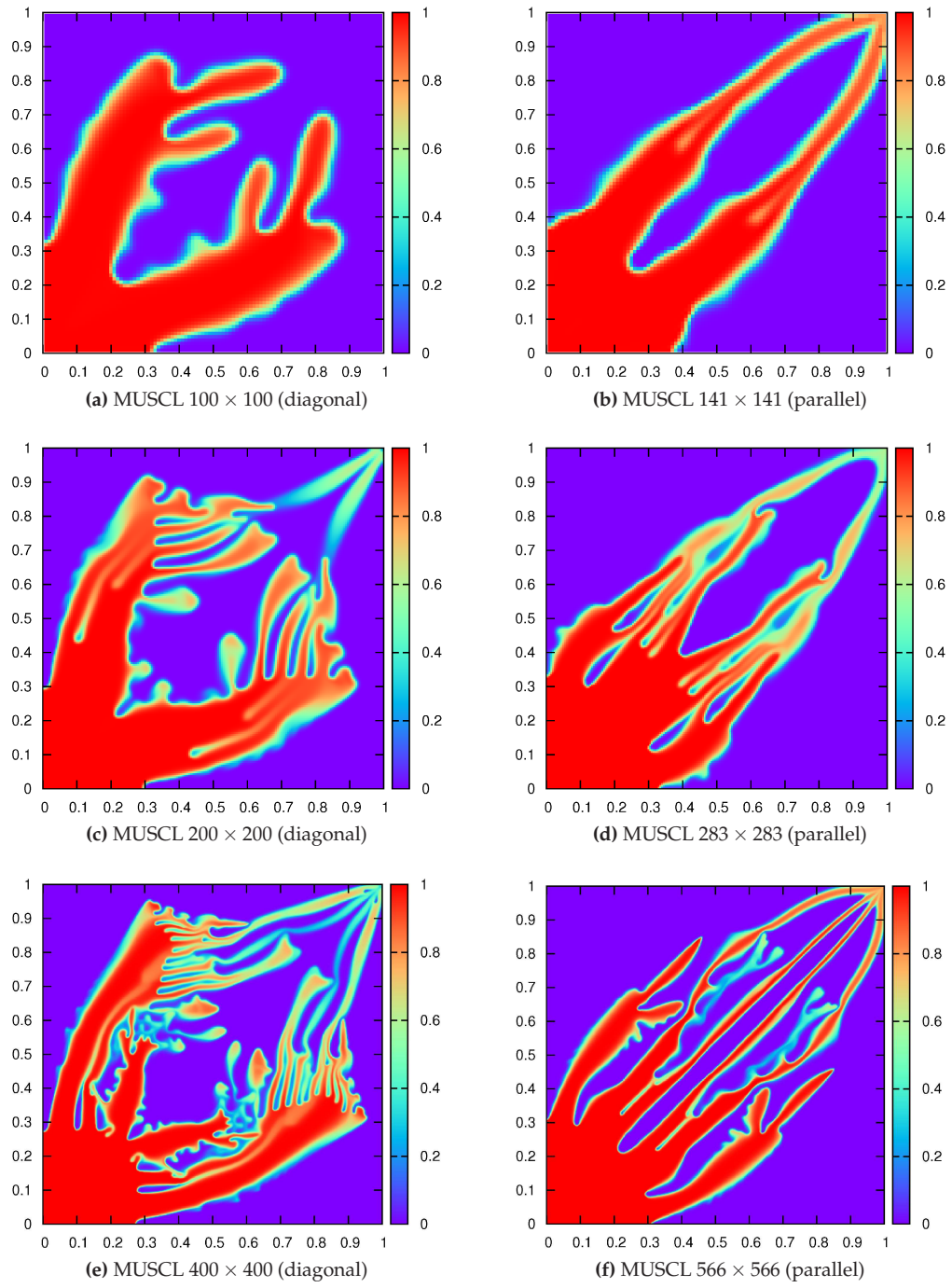


Figure 4.3: Results for the MUSCL scheme in the absence of dispersion ($Pe_l = Pe_t = \infty$); left column: diagonal solutions; right column: parallel solutions.

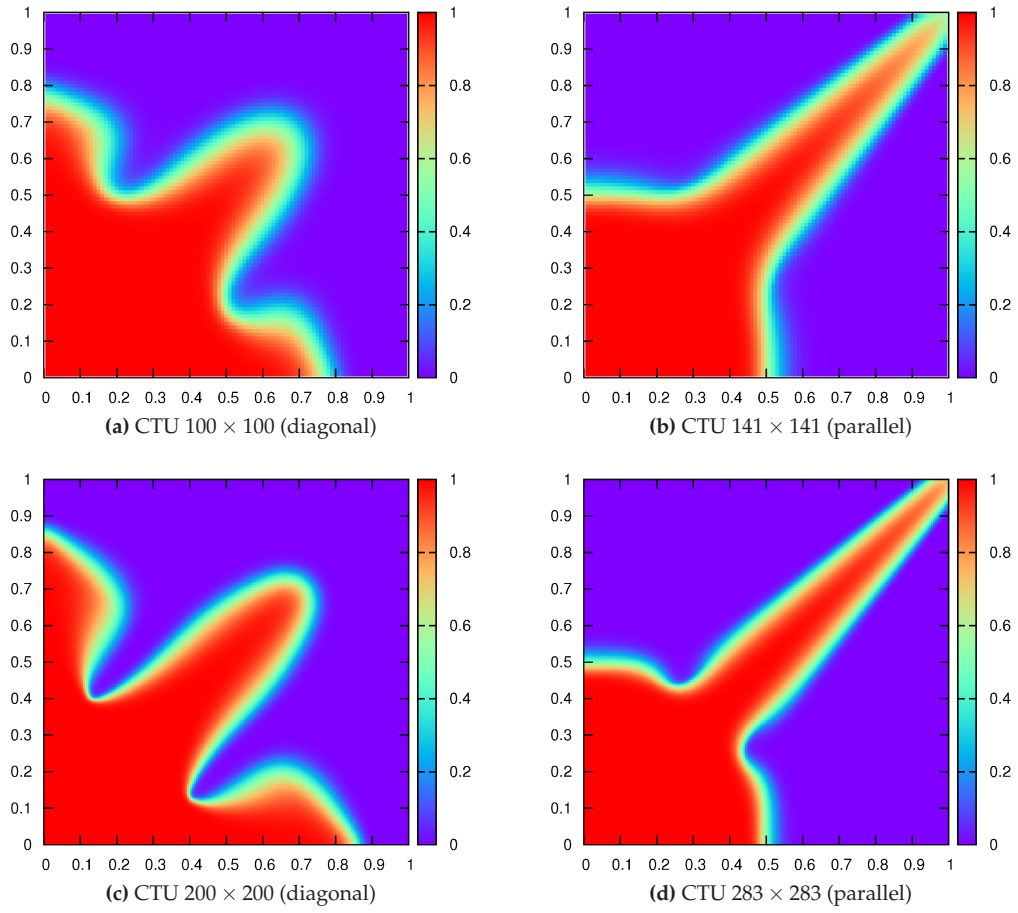


Figure 4.4: Results for the CTU scheme with the dispersion tensor \mathbf{D}^{SB} ; left column: diagonal solutions; right column: parallel solutions.

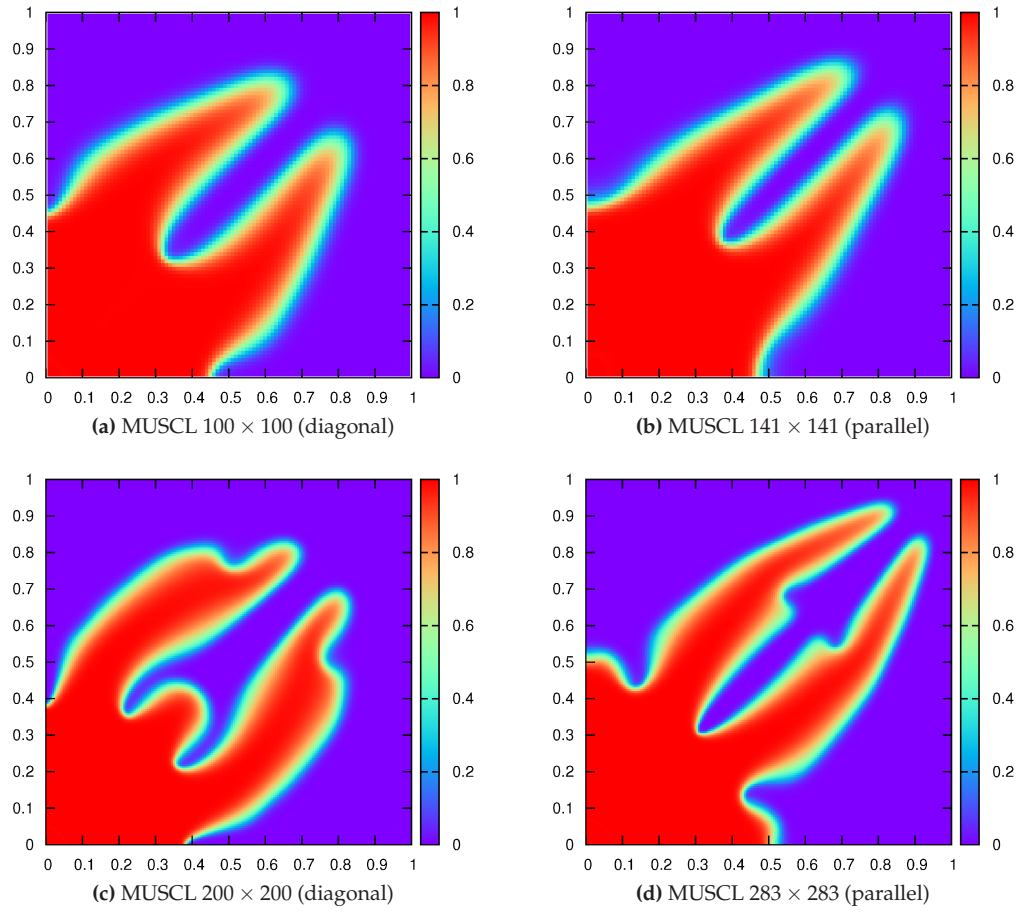


Figure 4.5: Results for the MUSCL scheme with the dispersion tensor \mathbf{D}^{SB} ; left column: diagonal solutions; right column: parallel solutions.

4.3.3 Modified numerical dispersion

The previous examples suggest that the magnitudes of numerical dispersion, regulated through α_l and α_t , are not high enough to suppress the effects of grid orientation in the range of resolutions we are interested in. Experimenting with different configurations, we designed a dispersion tensor, \mathbf{D}^M , that gives improved results for the MUSCL scheme. Our choice for the coefficients is $\alpha_l = \frac{3}{4}\Delta x$ and $\alpha_t = \frac{9}{4}\Delta x$ yielding

$$\mathbf{D}^M = \frac{3\Delta x}{4|\mathbf{u}|} \begin{pmatrix} u^2 + 3v^2 & -2uv \\ -2uv & v^2 + 3u^2 \end{pmatrix} \quad (4.3.2)$$

where all elements are still of order $\mathcal{O}(\Delta x)$ in space. These coefficients are motivated by trial and improvement, rather than analytical derivation or theory.

The results obtained for the CTU scheme show better agreement for low resolutions (Figs. 4.6a,b) when \mathbf{D}^M is applied. Furthermore, we can see that the concentration front is considerably smeared out. For higher resolutions, the choice of α_l and α_t turns out not to be suitable. Whereas the diagonal results do not change much (Figs. 4.6a,c), the parallel solutions (Figs. 4.6b,d) differ substantially.

The situation changes significantly for the MUSCL scheme. Figs. 4.7a,b show a surprisingly good agreement and the GOE is minimal as compared to the previous results. Doubling the resolution has a slight effect on the parallel result (Fig. 4.7d) but the deviation between the two configurations is still small. Further increasing the resolution to 400×400 and 566×566 cells, respectively, we can see that the effects of grid orientation are only hidden but not eliminated. The diagonal solution (Fig. 4.7e) changes its shape slightly, whereas the parallel (Fig. 4.7f) remains basically the same. In Fig. 4.11a, the results are compared for a single contour line. For the range of 100 – 200 and 141 – 283 cells per dimension, respectively, the two configurations agree well.

4.3.4 Errors for mass and velocity

Finite volume methods are by definition conservative, since each of the four fluxes used for updating a cell (i, j) in two spatial dimensions is used for updating a neighbouring cell as well. Special care must be taken at the domain boundary. Let us consider the upwind scheme as an easy example. If the velocity vanishes exactly on the boundary for zero gradient Neumann boundary conditions, all corresponding intercell fluxes \mathcal{F}^G and \mathcal{G}^G will vanish as well according to equation (3.6.18). It is therefore important that the velocity vanishes exactly at the interface. This is implicitly satisfied by our formulation of the pressure equation and the way we calculate the velocity. Thus the mass influx from

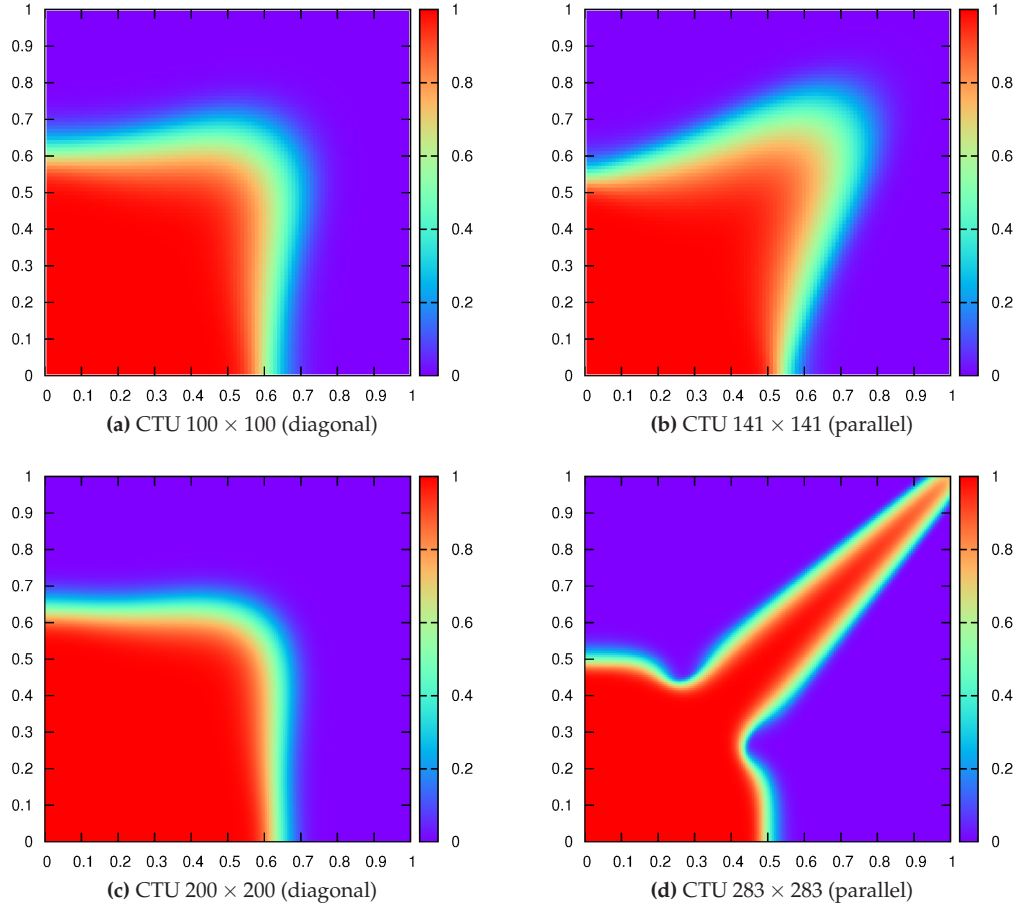


Figure 4.6: Results for the CTU scheme with the new dispersion tensor \mathbf{D}^M ; left column: diagonal solutions; right column: parallel solutions.

the boundary is zero and the mass loss is negligible due to the conservation property of the finite volume scheme (see Fig. 4.8). Let us denote the total injected mass by m_I , the produced mass by m_p , the mass in the system by m_S and the lost mass by Δm . We then calculate the relative mass error \mathcal{E}_m with respect to the total mass influx according to

$$\mathcal{E}_m := \frac{\Delta m}{m_I} = \left| \frac{m_S + m_O - m_I}{m_I} \right|. \quad (4.3.3)$$

The time evolution of the mass error is illustrated in Fig. 4.8. The absolute velocity error, i.e. the modulus of the velocity summed over the boundary, is identically zero for all the simulations carried out. The relative tolerance ϵ (see Tab. 4.1) chosen for the elliptic equation reflects itself in an error of the divergence of the velocity (Fig. 4.9). According to equation (2.1.17), the divergence of the velocity should be equal to the source term q

everywhere exactly. Fig. 4.9 shows the time evolution of the L_1 -divergence error defined by

$$L_1(\nabla \cdot \mathbf{u}) := \Delta x^2 \sum_{ij} \left| (\nabla \cdot \mathbf{u})_{ij} - q_{ij} \right| \quad (4.3.4)$$

for different values of ϵ , where $(\cdot)_{ij}$ denotes the numerical approximation at the centre of the cell (i, j) .

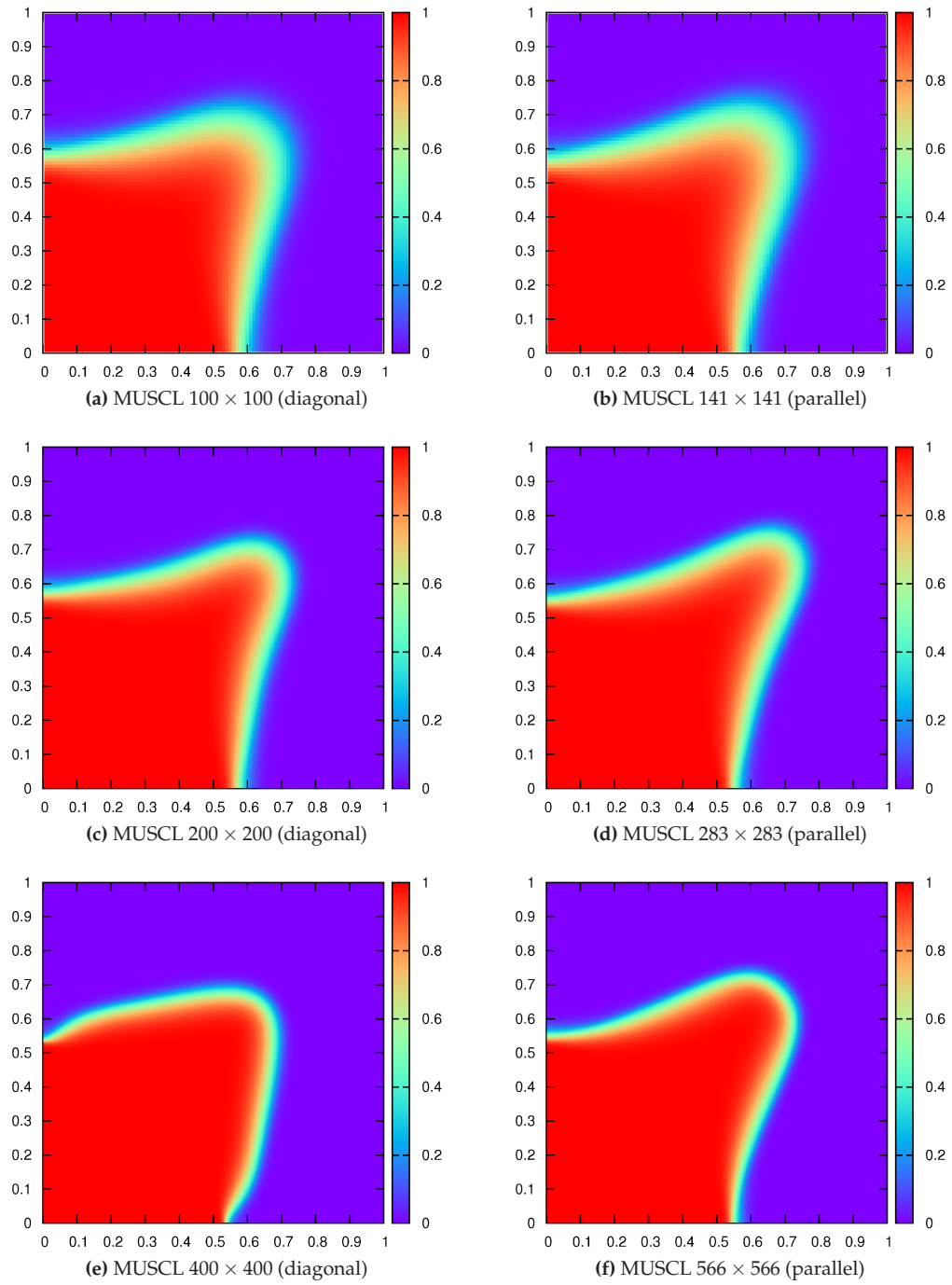


Figure 4.7: Results for the MUSCL scheme with the new dispersion tensor \mathbf{D}^M ; left column: diagonal solutions; right column: parallel solutions.

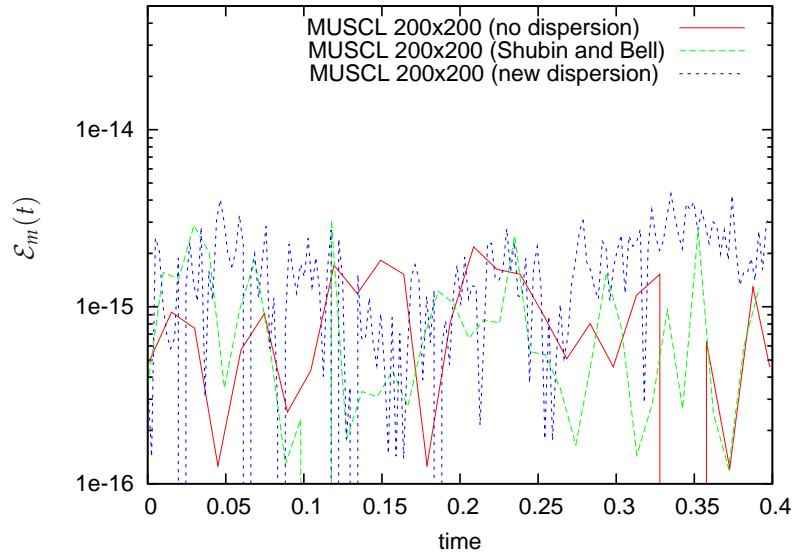


Figure 4.8: Relative mass error for the partial dispersion case; only every 40th time step is plotted for readability.

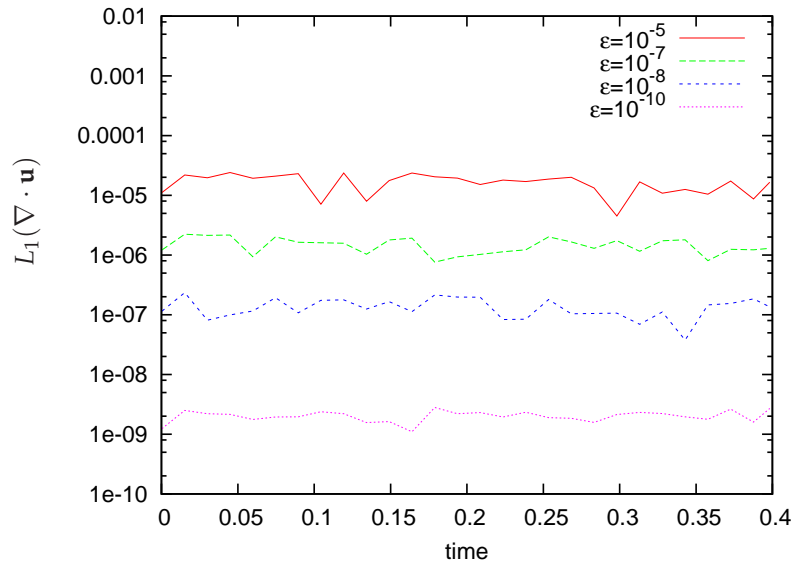


Figure 4.9: L_1 -error for the divergence of the velocity as a function of time for various values of ϵ ; settings: MUSCL scheme, $Pe_l = Pe_t = \infty$, 200×200 cells, diagonal configuration; only every 40th time step is plotted for readability.

4.4 PARTIAL DISPERSION STUDY

We can improve the GOE by applying dispersion uniformly over the whole domain. This approach has the disadvantage that the concentration front is smeared out considerably (see Figs 4.7a-d). Ideally, we want to provide the necessary stabilization and preserve the high concentration gradient at the front.

In this section we show that it might be sufficient to stabilize the flow by applying dispersion to the cells in the vicinity of the front only. The information about the time evolution of the front is given by a level set function which is advected with the same velocity field. For each time step, we can determine the zero crossing of the level set function and flag a certain range of neighbouring cells, defined by a parameter σ , with respect to the zero contour line. In order to do this we use a mask of the same dimensions as the concentration array, where each element is set to *false* initially. Then we determine all the cells, (i, j) , for which the level set function undergoes a zero crossing with respect to the neighbouring cells, $(k, l) \in [i - 1, i + 1] \times [j - 1, j + 1]$, and denote the set by \mathcal{M} . Subsequently, we update the cells $[i - \sigma, i + \sigma] \times [j - \sigma, j + \sigma]$ to *true* for each cell $(i, j) \in \mathcal{M}$. Finally, we apply dispersion only for the flagged cells in order to control the smearing of the front.

The results for this novel approach are presented in Fig. 4.10. The solid black line in Figs. 4.10 shows the initial front and the dashed line shows the contour Δx at the end of the simulation. The level set function is initialized with a circle so that the contour line Δx coincides with the injection well. Instead of tracking the level zero contour, we track the level Δx contour because it is favourable numerically. In the parallel configuration the level set function is initialized accordingly for the two parts of the domain separated by the line connecting the production wells.

The quality of the results with regards to the GOE is as good as in the uniform dispersion case. However, the new results are clearly improved with respect to front smearing (Figs. 4.12a,b). The position of the contour line 0.95 shows a stronger trend towards the production well in the partial dispersion case. The largest deviation is exhibited between the parallel results, which is still smaller than the largest deviation in the uniform dispersion case.

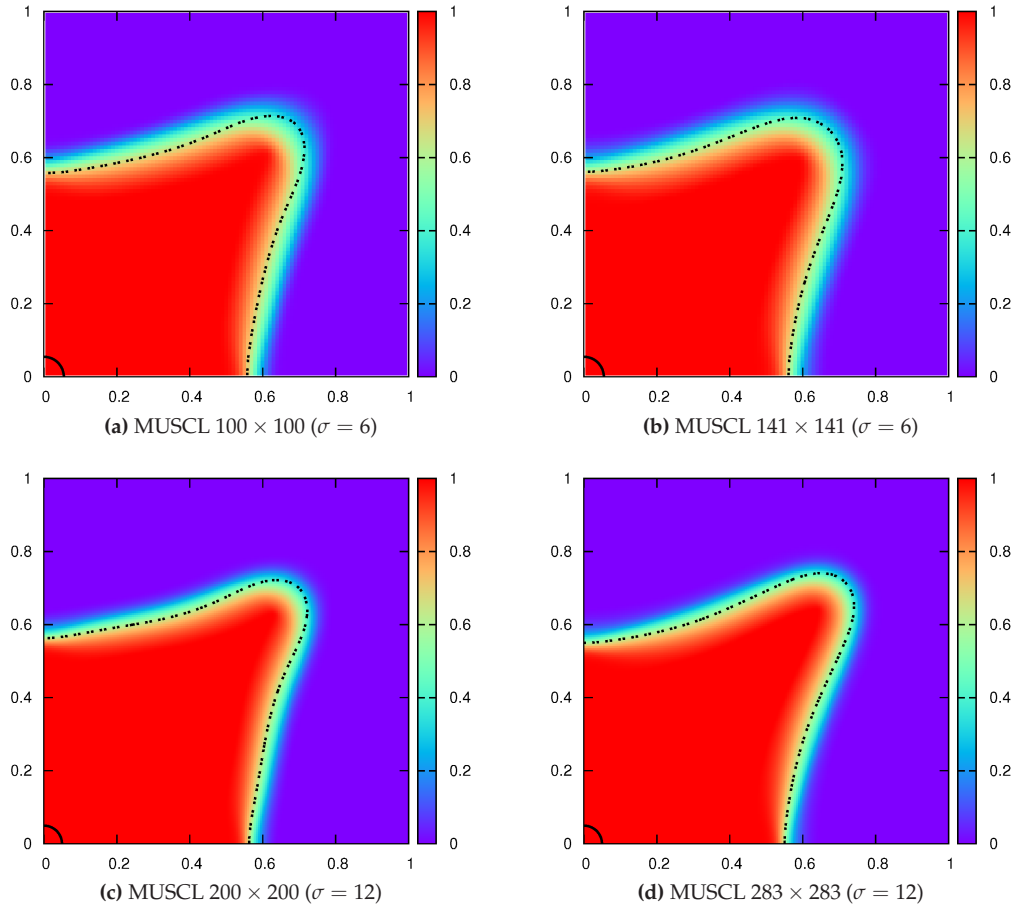


Figure 4.10: Results for the MUSCL scheme for partial dispersion.

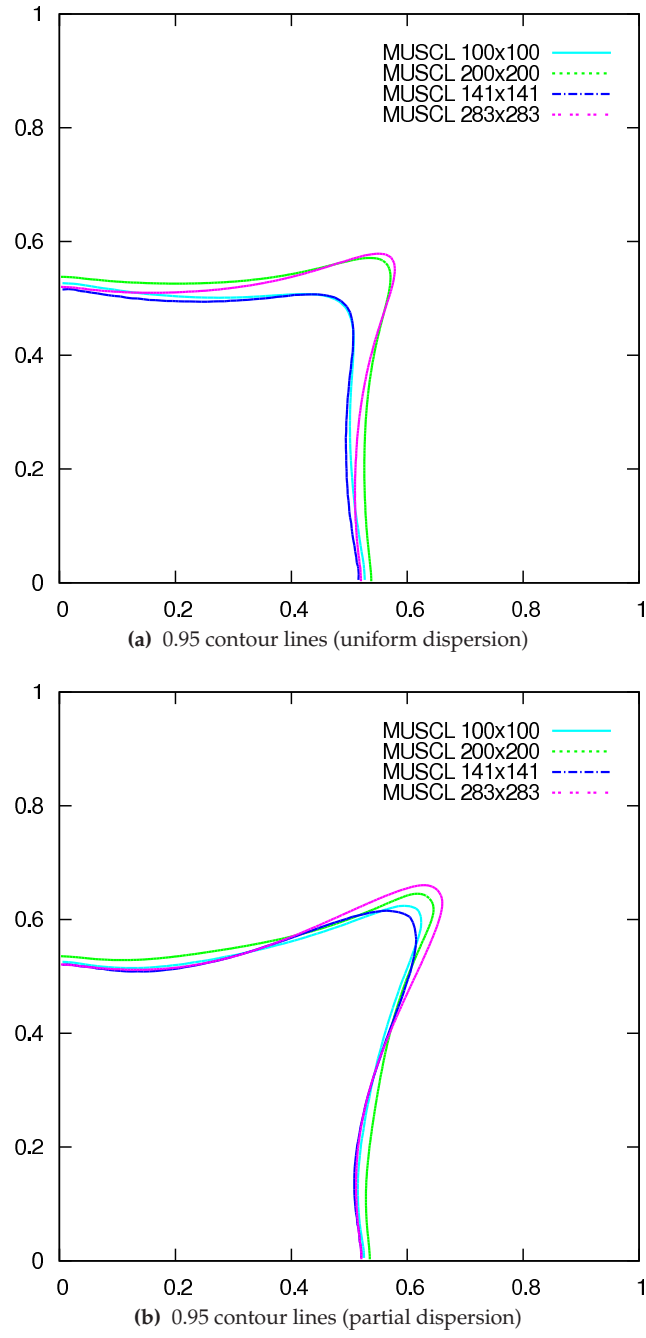
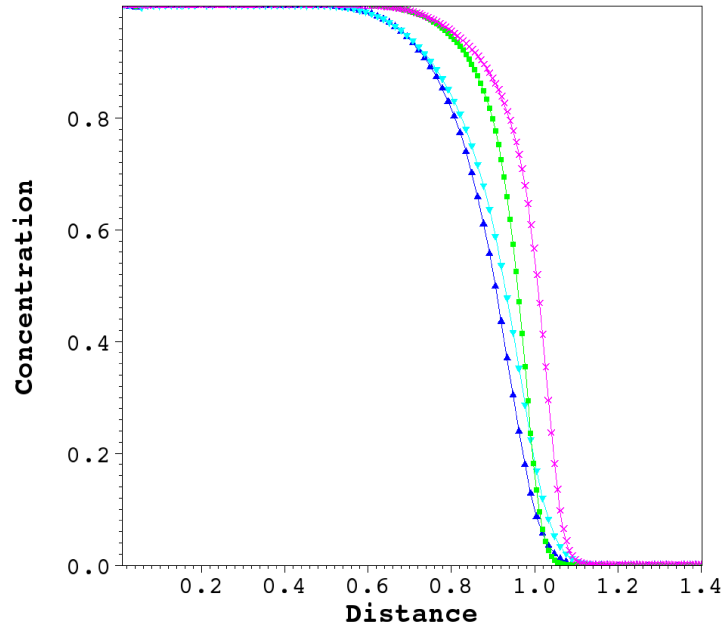
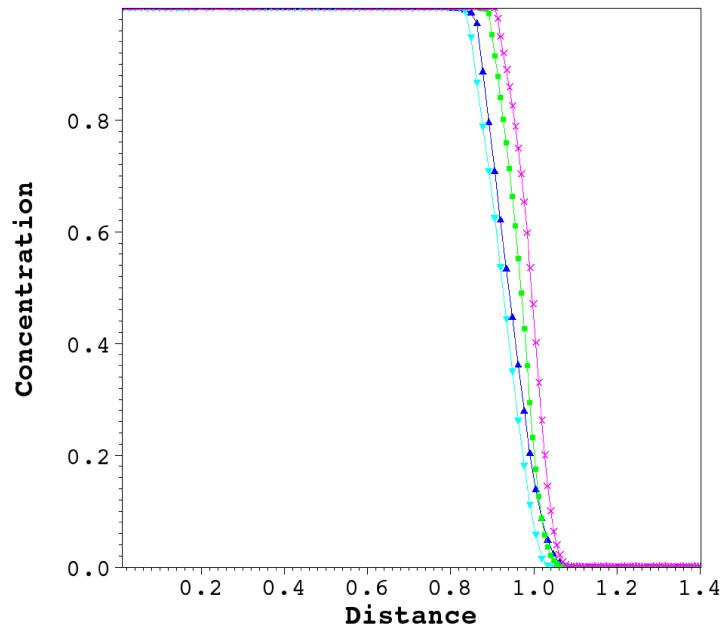


Figure 4.11: (a) Comparison of the position of the 0.95 contour line for the uniform dispersion case in Figs. 4.7a-d; (b) the same comparison for the partial dispersion study in Figs. 4.10a-d; $\sigma = 6$ for 100×100 (141×141) cells and $\sigma = 12$ for 200×200 (283×283) cells.



(a) uniform dispersion



(b) partial dispersion

Figure 4.12: Slices from the injection well to the production well: (a) for the results in Figs. 4.7a-d; (b) for the results in Figs. 4.10a-d; the same color scheme is used as in Fig. 4.11.

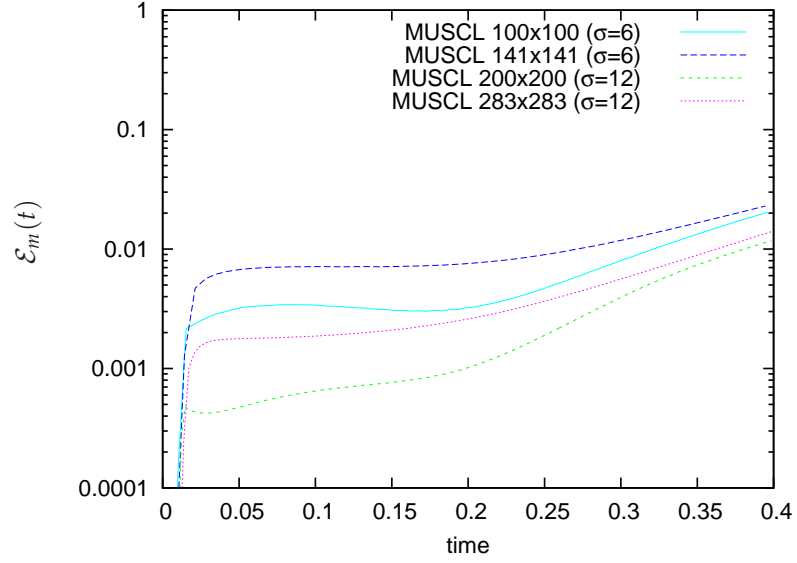


Figure 4.13: Relative mass error with respect to the mass influx as a function of time for the results in Figs. 4.10a-d; only every 40th time step is plotted for readability.

We expect the mass error to rise considerably if we only modify flagged cells (see Fig. 4.13). If dispersion is used for cell (i, j) , the concentration will change for that cell because fluid goes to or comes from neighbouring cells. However if we do not update the neighbouring cells as well, we get local sources and sinks and hence a high mass error. In future work, this could be addressed by employing a signed distance level set function and modulating the dispersion term with respect to the distance to the front. Additionally, some kind of boundary correction between flagged and unflagged cells could be applied.

Conclusions

Recent simulations of thermal flow in porous media exhibit a dependency of the result on the underlying grid. In this report, we investigated the grid orientation effect exhibited by various finite volume schemes as applied to the equations of miscible displacement in the quarter five-spot configuration. These equations are simpler than those used for thermal flow in porous media and hence easier to study. The schemes under investigation were chosen with respect to their applicability to simulations of thermal flow in porous media.

The effects of grid orientation strongly depend on the spatial resolution. Even for low resolutions, the numerical diffusion-dispersion of the scheme is not sufficient anymore to stabilize the front. Numerical errors can trigger instabilities that cause different results. Usually, a numerical stabilization term is employed to hide the asymmetries in the spatial discretization. This term is proportional to the spatial resolution and its stabilization effect decreases as the grid is refined. In previous work, only low-resolution results of less than 100×100 cells were considered for a comparison of the two symmetric configurations of the problem.

We investigated resolutions of up to 566×566 cells and suggested a new numerical dispersion tensor for the MUSCL-Hancock scheme. This tensor is a modification of the rotationally invariant dispersion tensor previously used by Shubin and Bell [29]. Employing the new stabilization term, we were able to minimize the GOE for resolutions between 100×100 (141×141) and 200×200 (283×283) cells. In addition, we showed that the effect becomes worse as we increase the resolution further.

High amounts of numerical dispersion were necessary for the range of resolutions under investigation causing a strong smearing of the concentration front. In this report, we developed a new scheme based on a level set function which addresses this problem in a novel way. We suggest applying dispersion only in the vicinity of the front, instead

of applying it uniformly, in order to reduce smearing whilst providing the necessary stabilization. The information about the time evolution of the front is provided by a level set function. The presented numerical results show little dependence on the grid and we could demonstrate that smearing is reduced considerably as compared to the results for uniform dispersion.

5.1 FUTURE WORK

In future we want to address the problem of grid orientation in simulations of thermal flow in porous media. The results in this report are promising and we are confident that the same methodology is applicable to the more complex system.

With regard to the mass error, we found that the scheme we employ for the parabolic equation can be cast into a finite volume form. Applying dispersion to a cell will lead to nonzero intercell fluxes for that cell. If we were to update the neighbouring cells in a conservative manner, they would get a contribution from that cell even in the absence of dispersion. Thus, this could serve as a natural boundary correction between cells where numerical dispersion is applied and unmodified cells.

Alternatively, a continuous approach for reducing the front smearing could be achieved by restoring the signed distance information of the level set function. This is usually done by solving an Eikonal equation after the update step. The numerical dispersion could then be modulated using the distance to the interface, e.g. with a gaussian curve. We would expect similar improvements for this approach as for the partial dispersion case, if the modulating function decreases fast away from the interface.

In this work we applied a numerical dispersion tensor, where the longitudinal and transverse magnitudes were proportional to the grid spacing. In future work, we will also consider a fixed dispersion under high resolution, so that the term can be well resolved. More advanced methods, such as *adaptive mesh refinement* (AMR) in combination with a parallel *multigrid solver* for the pressure equation, could help us simulate at significantly higher resolutions than are seen anywhere else in the literature ($> 5000 \times 5000$ cells). From this we hope to get new insights of how the GOE could behave once we are able to resolve diffusion-dispersion closer to physical scales.

Bibliography

- [1] C. Badura. C++ solvers for sparse systems of linear equations. http://aam.mathematik.uni-freiburg.de/IAM/Research/projectskr/lin_solver, 1998. (last access: 16/08/2012).
- [2] J.B. Bell, C.N. Dawson, and G.R. Shubin. An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74(1):1–24, January 1988.
- [3] C.W. Brand, J.E. Heinemann, and K. Aziz. The grid orientation effect in reservoir simulation. *SPE Symposium on Reservoir Simulation*, 1991.
- [4] A. Briginshaw. Elementary numerical analysis. Technical report, University of Cambridge, 2011.
- [5] C.-Y. Chen and E. Meiburg. Miscible porous media displacements in the quarter five-spot configuration. Part 1. The homogeneous case. *Journal of Fluid Mechanics*, 371:233–268, 1998.
- [6] W.H. Chen, L.J. Durlofsky, B. Engquist, and S. Osher. Minimization of grid orientation effects through use of higher order finite difference methods. *SPE Advanced Technology Series*, 1(2):43–52, 1993.
- [7] M.A. Christie and D.J. Bond. Multidimensional flux-corrected transport for reservoir simulation. *SPE Reservoir Simulation Symposium*, 1985.
- [8] M.A. Christie and D.J. Bond. Detailed simulation of unstable processes in miscible flooding. *SPE Reservoir Engineering*, 2(4):514–522, 1987.
- [9] C. Curtis, R. Kopper, E. Decoster, A. Guzmán-García, C. Huggins, L. Knauer, M. Minner, N. Kupsch, L.M. Linares, H. Rough, and M. Waite. Heavy-oil reservoirs. *Oilfield Review*, 14(3), 2002.

- [10] B.L. Darlow, R.E. Ewing, and M.F. Wheeler. Mixed finite element method for miscible displacement problems in porous media. *SPE Journal*, 24(4):391–398, 1984.
- [11] M.G. Edwards. Cross flow tensors and finite volume approximation with by deferred correction. *Computer Methods in Applied Mechanics and Engineering*, 151(1-2):143–161, 1998.
- [12] R.E. Ewing. *The mathematics of reservoir simulation*. SIAM, Philadelphia, 1983.
- [13] R.E. Ewing, T.F. Russell, and L.C. Young. An anisotropic coarse-grid dispersion model of heterogeneity and viscous fingering in five-spot miscible displacement that matches experiments and fine-grid simulations. *SPE Symposium on Reservoir Simulation*, February 1989.
- [14] R.P. Fedkiw, T.D. Aslam, B. Merriman, and S. Osher. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999.
- [15] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.
- [16] S.C.M. Ko and A.D.K. Au. A weighted nine-point finite-difference scheme for eliminating the grid orientation effect in numerical reservoir simulation. *SPE Annual Technical Conference and Exhibition*, 1979.
- [17] J.E. Kozdon. *Numerical methods with reduced grid dependency for enhanced oil recovery*. PhD thesis, Stanford University, 2009.
- [18] J.E. Kozdon, B. Mallison, and M. Gerritsen. Robust multi-d transport schemes with reduced grid orientation effects. *Transport in Porous Media*, 78(1):47–75, October 2009.
- [19] S. Lamine and M.G. Edwards. Multidimensional upwind convection schemes for flow in porous media on structured and unstructured quadrilateral grids. *Journal of Computational and Applied Mathematics*, 234(7):2106–2117, August 2010.
- [20] R.J. LeVeque. *Numerical methods for conservation laws*. Birkhäuser, Basel, 1992.
- [21] R.J. LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *Journal of Computational Physics*, 131(2):327–353, 1997.
- [22] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge University Press, New York, 2002.

- [23] J.M. McDonough. Lectures in computational fluid dynamics of incompressible flow: Mathematics, algorithms and implementations. Technical report, Departments of Mechanical Engineering and Mathematics, University of Kentucky, 2007.
- [24] F.B.J. Monmont, D.E.A. van Odyck, and N. Nikiforakis. Experimental and theoretical study of the combustion of n-triacontane in porous media. *Fuel*, 93:28–36, March 2012.
- [25] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer-Verlag, New York, 2003.
- [26] P. Petitjeans, C.-Y. Chen, E. Meiburg, and T. Maxworthy. Miscible quarter five-spot displacements in a Hele-Shaw cell and the role of flow-induced dispersion. *Physics of Fluids*, 11(7):1705–1716, 1999.
- [27] T. Potempa. An improved implementation of the McCracken and Yanosik nine point finite difference procedure. *Applied Numerical Mathematics*, 1(2):261–272, 1985.
- [28] G. Shiralkar. Reservoir simulation of generally anisotropic systems. *SPE Reservoir Engineering*, 5(3):409–414, 1990.
- [29] G.R. Shubin and J.B. Bell. An analysis of the grid orientation effect in numerical simulation of miscible displacement. *Computer Methods in Applied Mechanics and Engineering*, 47(1-2):47–71, 1984.
- [30] C.T. Tan and G.M. Homsy. Stability of miscible displacements in porous media: Radial source flow. *Physics of Fluids*, 30(5):1239–1245, 1987.
- [31] M.R. Todd, P.M. O’Dell, and G.J. Hirasaki. Methods for increased accuracy in numerical reservoir simulators. *SPE Journal*, 12(6):515–530, 1972.
- [32] E.F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, 2009.
- [33] D.E.A. van Odyck, J.B. Bell, F.B.J. Monmont, and N. Nikiforakis. The mathematical structure of multiphase thermal models of flow in porous media. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465(2102):523–549, February 2009.
- [34] D.E.A. van Odyck, S. Lovett, F. Monmont, and N. Nikiforakis. An efficient shock capturing scheme for multicomponent multiphase thermal flow in porous media. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, July 2012.

- [35] R. Wang and R.J. Spiteri. Linear instability of the fifth-order WENO method. *SIAM Journal on Numerical Analysis*, 45(5):1871–1901, 2007.
- [36] J.L. Yanosik and T.A. McCracken. A nine-point, finite-difference reservoir simulator for realistic prediction of adverse mobility ratio displacements. *SPE Journal*, 19(4):253–262, August 1979.
- [37] H.R. Zhang, K.S. Sorbie, and N.B. Tsibuklis. Viscous fingering in five-spot experimental porous media: New experimental results and numerical simulation. *Chemical Engineering Science*, 52(1):37–54, January 1997.