Imperial College Consortium on Pore-Scale Modelling

Yearly progress and final report

January 3rd 2011

Martin Blunt, Branko Bijeljic, Peyman Mostaghimi and Ali Raeini
Executive Summary

This is the first annual report of the new phase of the Imperial College Consortium on Pore-Scale Modelling. In our project meeting we will highlight the progress we have made over the last year as well as presenting plans for the future. We have returned to fundamentals to study flow and transport directly in pore-space images, embarked on a study of reactive transport and complement our modelling through an extensive experimental programme, including imaging of multiphase fluid distributions at reservoir conditions at the pore scale.

The first chapter describes the results of experiments that measured the residual saturation in three water-wet sandstones as a function of initial non-wetting phase saturation. This is part of a separate ongoing project sponsored by Shell and Qatar Petroleum and the Qatar Science and Technology Park. The results are then compared with predictions of pore-scale modelling using networks extracted from micro-CT images of the rocks of interest. The good comparison between measurement and theory suggests that we can capture the processes that determine trapping at the pore scale in consolidated rocks. This work will continue to study carbonates, rocks of different wettability and the trapping of carbon dioxide at super-critical (reservoir) conditions. This work was presented at the SPE Annual Meeting in Florence in September.

The second chapter describes the work performed by Peyman Mostaghimi, a PhD student funded by the consortium. He has developed an efficient code to solve for Stokes (slow) flow directly on pore-space images. In addition, a new streamline algorithm is used to simulate advection – this is similar to the method used in reservoir simulation for grid blocks with no solid boundaries, but uses different semi-analytical approximations for the flow field within blocks with solid faces. Coupled with a random walk method to account for diffusion, an elegant method is developed to simulate dispersion that gives good prediction of measured dispersion coefficients as a function of Peclet number. Again this work was presented at the SPE Annual Meeting in Florence. Future work will study transport in highly heterogeneous systems (such as carbonates), pres-asymptotic dispersion and reactive transport.

The third chapter presents a detailed review of different methods that can be employed to simulate multiphase flow on pore-space images. A volume of fluid method with interface tracking is selected for further development and initial results are shown that demonstrate the potential of the method to simulate displacement at low capillary number without the need to extract networks. This chapter forms the first-year transfer report for Ali Raeini, another PhD student funded by the project. The research holds considerable promise of a real breakthrough in our efforts to capture the effects of complex geometry, multiple phases and wettability at the pore scale. Of course, another popular method for the simulation of multiphase flow is the lattice Boltzmann technique, and our latest results using this approach will be presented by Edo Boek at the project meeting.
The current researchers in the group are:

Martin Blunt, Professor of Petroleum Engineering – overall supervision and fundamental studies of three-phase flow.

Branko Bijeljic, Research Fellow – dispersion and reactive transport in porous media.

Edo Boek, Senior Lecturer in Chemical Engineering – fundamentals of flow in porous media and wettability.

Rafi Blumenfeld, Research Fellow – statistical analysis of granular packs.

Peyman Mostaghimi – 3rd year PhD student – Stokes flow and transport in pore-space images.

Ali Raeini – 2nd year PhD student – Multiphase flow on pore-space images.

All our publications, theses, reports and presentations are available on our Website: http://www3.imperial.ac.uk/earthscienceandengineering/research/perm/porescalemodelling

Project publications in 2010

Journal publications

Conference proceedings

Our current sponsors are: BG, BP, JOGMEC, Shell, Statoil and Total. I would like to thank you all for your continued support that allows us to fund so many students and without which this research would not be possible.

Martin Blunt, London, January 2011
Chapter 1

SPE 133798

Capillary Trapping in Water-Wet Sandstones: Coreflooding Experiments and Pore-Network Modeling
Christopher H. Pentland, Yukie Tanino, Stefan Iglauer, Martin J. Blunt, SPE, Imperial College London

This paper was prepared for presentation at the SPE Annual Technical Conference and Exhibition held in Florence, Italy, 19–22 September 2010.

Abstract
Displacement experiments using the porous plate method were conducted on water-wet sandstones to measure the capillary trapping of oil by waterflooding as a function of its saturation after primary drainage. Three sandstone samples ranging in porosity from 12.2% to 22.1% were considered. Experiments on two samples were conducted at an elevated temperature and back-pressure of 343K and 9MPa respectively; experiments on the third sample were conducted at ambient conditions (292 to 297K and 0.06 to 0.17MPa). Residual oil saturation increases monotonically, but with a decreasing gradient, as initial saturation increases.

The dependence of residual saturation on initial saturation is accurately predicted by a two-phase pore-network simulator when a uniform distribution of intrinsic contact angles between 35° and 65° is assumed. The networks were extracted from X-ray microtomography images of small samples of the same rock as those used in the experiments. The laboratory measurements are also accurately described by trapping models proposed by Land (1968) and Spiteri et al. (2008). The residual saturations we measured were higher than in previous displacement experiments, suggesting, for example, that capillary trapping may be an effective way to store substantial quantities of carbon dioxide in aquifers.

Introduction
The trapping of non-wetting phase in a porous medium as discontinuous pore-scale droplets by capillary forces, or capillary trapping, has been studied extensively because of its importance to oil recovery and contaminant remediation. In these applications, the motivation is the extraction of the trapped phase. In contrast, in the context of geological carbon storage, the objective is to maximize trapping. Carbon dioxide (CO₂) is injected into a target geological formation, forming a continuous plume. As the injected CO₂ is driven upwards by buoyancy, ambient groundwater will flow into its wake to replace it. This replacement of CO₂ by groundwater behind the rising CO₂ plume can be regarded as a re-imbibition process. Accordingly, a portion of the migrating CO₂ will be rendered immobile within the pores of the rock by capillary forces, and will no longer be at risk of leakage to the atmosphere provided that local conditions remain unchanged. Further, this process can be enhanced through injection of additional brine (Juanes et al. 2006; Qi et al. 2009).

While many studies report laboratory measurements of capillary trapping in porous media (e.g., Abrams 1975; Agbalaka et al. 2009; Chatzis and Morrow 1984), literature that investigate the dependence of capillary trapping on initial non-wetting phase saturations are more limited. If we further restrict our consideration to experiments in which drainage and imbibition were achieved by displacement of phases within the porous medium, which is the process relevant to carbon storage, and to experiments on uniformly-wet porous media, existing data are limited to those listed in Table 1; a more extensive review of the literature is provided by Pentland et al. (2010). This paper complements these data with new oil/brine coreflooding experiments at flow rates representative of field conditions.

Residual saturation data have traditionally been compared to a number of empirical trapping models. In this paper, our data will be compared to two such models. The first, which is the most widely used in the literature and commercial simulators, was proposed by Land (1968):

\[ S_{\text{res}} = \frac{S_{\text{ini}}}{1 + S_{\text{ini}} \left( \frac{1}{S_{\text{res}}} - \frac{1}{1 - S_{\text{ini}}} \right)} \]  

(1)
where $S_{nwi}$ and $S_{nwr}$ are the non-wetting phase saturation after primary drainage and after secondary imbibition, respectively, $S_{nwi}^{\text{max}}$ is the maximum $S_{nwi}$, and $S_{wc}$ is the connate wetting phase saturation. The second, a quadratic relationship between initial and residual saturation, was proposed by Spiteri et al. (2008):

$$S_{nwi} = \alpha S_{nwi}^{\text{max}} - \beta S_{wc}^2,$$

where $\alpha$ and $\beta$ are fitting constants. Spiteri et al. (2008) report $\alpha$ and $\beta$ that best-fit their pore-scale simulations of a Berea sandstone using macroscopic intrinsic contact angles ranging from $\theta = 20^\circ$ to $160^\circ$.

<table>
<thead>
<tr>
<th>Source</th>
<th>phases</th>
<th>porous media</th>
<th>porosity</th>
<th>permeability [m$^2$]</th>
<th>symbol in Fig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kleppe et al. 1997</td>
<td>air/oil</td>
<td>artificial consolidated medium</td>
<td>0.43</td>
<td>4.7×10^{-12}</td>
<td>grey —</td>
</tr>
<tr>
<td>Al Mansoori et al. 2010</td>
<td>air/brine</td>
<td>sand pack</td>
<td>0.37 ± 0.002</td>
<td>(320±3)×10^{-13}</td>
<td>green +</td>
</tr>
<tr>
<td>Pentland et al. 2010</td>
<td>oil/brine</td>
<td>sand pack</td>
<td>0.37 ± 0.002</td>
<td>(320±3)×10^{-13}</td>
<td>green ×</td>
</tr>
<tr>
<td>Pickell et al. 1966</td>
<td>mercury/air; oil/water</td>
<td>Dalton sandstone</td>
<td>0.286</td>
<td>4.17×10^{-13}</td>
<td>grey ○</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.248</td>
<td>1.90×10^{-13}</td>
<td>grey □</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.291</td>
<td>4.14×10^{-13}</td>
<td>grey ◊</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.286</td>
<td>2.55×10^{-14}</td>
<td>grey △</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.28</td>
<td>2.07×10^{-13}</td>
<td>grey △</td>
</tr>
</tbody>
</table>

Table 1: Studies that report residual saturation as a function of initial non-wetting phase saturation. Only studies that achieved drainage and imbibition by displacement are listed.

Displacement experiments were conducted to measure the capillary trapping of oil by waterflooding as a function of its saturation after primary drainage in three water-wet sandstone cores. The results are compared with the two empirical models given above, and with simulations of the same sequence of displacements using the two-phase flow pore-network model developed by Valvatne and Blunt (2004) with networks previously extracted from X-ray microtomography images of samples from the same source rock as the cores.

**Experimental Procedure**

Three sandstone samples with porosities 0.221 (Berea), 0.122 (Clashach), and 0.156 (Stainton) were considered. Basic rock properties are summarized in Table 2. All three rocks are considered to be strongly water-wet. All cores were 3.8cm (1.5in.) in diameter and 7.5 to 7.7cm (3.0in.) in length. Experiments were conducted at ambient conditions in the Stainton core and at elevated temperature and pressure (ETP) in Clashach and Berea cores. In the ambient-condition runs, the temperature varied between 292 and 297K (19 to 24°C) and a back-pressure of 0.06 to 0.17MPa was applied. The ETP experiments were conducted inside an air bath at 343K (70°C) with a back-pressure of 9MPa.

The non-wetting phase was n-decane in the ETP experiments and n-octane in the ambient-condition experiments. This use of different non-wetting liquids is justified, as the estimated interfacial tensions of n-octane/water and n-decane/water at the relevant experimental conditions only differ by 7% (Table 2). The wetting phase was aqueous 5wt.%-sodium chloride, 1wt.%-potassium chloride synthetic brine. In each ambient-condition run, the density of the fluids was measured at 293.2K using a digital density-meter (Anton Paar DMA48). The average density of brine and n-octane across these runs were 1042kg m-3 and 702.7kg m-3, respectively. The density of the brine and n-decane were not measured at ETP, but are expected to be approximately 1030kg m-3 (Rogers et al. 1982) and 700kg m-3 (Banipal et al. 1991; Lee and Ellington 1965).
Table 2: Petrophysical properties of the core samples and fluid properties used in pore-network simulations.

<table>
<thead>
<tr>
<th>Property</th>
<th>Berea</th>
<th>Clashach</th>
<th>Stainton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effective porosity (± standard error)</td>
<td>0.221 ± 0.001</td>
<td>0.122 ± 0.003</td>
<td>0.156 ± 0.001</td>
</tr>
<tr>
<td>Brine permeability [m²]</td>
<td>5×10⁻¹³</td>
<td>8×10⁻¹⁴</td>
<td>3.8×10⁻¹⁴</td>
</tr>
<tr>
<td>Wetting phase</td>
<td>5wt.% NaCl, 1wt.% KCl aqueous solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-wetting phase</td>
<td>n-decane</td>
<td>n-octane</td>
<td></td>
</tr>
<tr>
<td>Interfacial tension [mN m⁻¹]</td>
<td>48.3ᵇ</td>
<td>51.64±0.04ᶜ</td>
<td></td>
</tr>
<tr>
<td>Wetting phase viscosity [Pa s]</td>
<td>4.554×10⁻⁴ᵈ</td>
<td>1.0903×10⁻³ᵉ</td>
<td></td>
</tr>
<tr>
<td>Non-wetting phase viscosity [Pa s]</td>
<td>5.47×10⁻⁴ᶠ</td>
<td>5.08×10⁻⁴ᵍ</td>
<td></td>
</tr>
</tbody>
</table>

*a Vinogradov et al. 2010.
*b Zeppieri et al. 2001; linearly extrapolated to 343K.
*c Zeppieri et al. 2001; 293.2 ± 0.1K.
*d Kestin, Khalia, and Correia 1981; 5.8wt.% NaCl aqueous solution at 10.0MPa, 343.2K.
*e Kestin, Khalia, and Correia 1981; 5.8wt.% NaCl aqueous solution at 100kPa, 293.2K.
*f Lee and Ellington 1965; 9480kPa, 344K.
*g CRC Handbook of Chemistry and Physics; 298K.

Coreflooding experiments. All experiments were conducted in custom-made horizontal Hassler-type core holders (radial confining pressure). A high-precision syringe pump (Teledyne ISCO 1000D or 500D) was used for all wetting and non-wetting fluid displacements. The capillary number was maintained at or below 3×10⁻⁷ during all displacement steps. All cores were cleaned before use by standard Soxhlet extraction with a mixture of methanol and toluene, oven-dried, and weighed.

Initial water saturation. The test core was saturated with de-gassed brine at experimental conditions inside the core holder. A minimum of five pore volumes of brine were injected through the core against back pressure. In ambient-condition runs, the core was subsequently removed from the Hassler cell, weighed, and then reinserted into the cell. The Hassler cell, containing the brine-saturated core, was weighed. The pore volume was determined from the increase in mass of the core in ambient-condition runs, and from the increase in mass of the Hassler cell in ETP runs.

Primary drainage. The porous plate method, in which a water-wet disk placed immediately downstream of the core retains the non-wetting phase inside the core, was used during primary drainage. With this method, equilibrium corresponds to uniform pressure distribution across the core in each of the phases, with the difference in the equilibrium pressure of the two phases corresponding to the capillary pressure. Capillary end effects are eliminated, and phase saturations are assumed to be uniform across the length of the core.

The non-wetting phase was injected into the core against a back-pressure using one of two methods: at constant pressure or at constant flow rate. The former was used for all experiments with Stainton, for drainage at a capillary pressure $P_c > 1000kPa$ in Berea, and for drainage at 50kPa < $P_c < 500kPa$ in Clashach. Here, equilibrium was considered achieved when brine production ceased and the volumetric rate of oil injection by the pump reached a constant (the leakage rate). Capillary pressure, and hence initial saturation, was varied by changing the injection pressure. For all other runs at ETP, a pre-determined volume of oil was injected at a constant flow rate corresponding to capillary numbers between 1.6×10⁻¹⁰ and 3.3×10⁻¹⁰. Once the volume necessary to achieve the target $S_{ni}^{wi}$ was injected, the system was allowed to equilibrate, and capillary pressure was measured. The duration of primary drainage ranged from 23 to 50 hours for Berea, 22 to 70 hours for Clashach, and 153 to 182 hours for Stainton. $S_{ni}^{wi}$ was determined from the volume of the injected non-wetting phase, the decrease in the mass of the Hassler cell containing the core from its brine-saturated state prior to oil injection and, for ambient-condition runs additionally, the volume and the mass of the effluent brine.

Waterflooding. Subsequently, the core was flooded with brine in the opposite direction as the oil injection. A minimum of five pore volumes of brine were injected in each ETP run; a minimum of ten pore volumes were injected in the ambient-condition runs. Residual saturation was determined from the decrease in the mass of the Hassler cell containing the core from its brine-saturated state for both ambient-condition and ETP runs, and, in addition, from the increase in mass of the Hassler cell during secondary imbibitions for ETP runs and from the decrease in the mass of the core relative to its dry state for ambient condition runs.

January 2011 6
Pore-network simulations. A corresponding set of simulations of primary drainage and waterflooding were performed using the two-phase flow pore-network simulator developed by Valvatne and Blunt (2004). Input parameters were matched with experimental conditions where possible. We used networks previously extracted (Table 3), using Dong and Blunt (2009)’s algorithm, from X-ray microtomography scans of samples from the same source as the cores in which the laboratory experiments were conducted (Fig. 1). The voxel resolution of the scans was 5.789µm × 5.789µm × 5.789µm for Berea and Clashach and 17.578µm × 17.578µm × 17.578µm for Stainton. Interfacial tension, brine viscosity, and oil viscosity were matched with experimental conditions, as listed in Table 2. A receding contact angle of 0° was assumed everywhere in the domain to describe a strongly water-wet state during primary drainage. For waterflooding, two values for the macroscopic intrinsic contact angle were considered: θ = 15° and 50°. The local intrinsic contact angle was randomly assigned, with equal probability, a value in the range θ ± 15°. Saturations and relative permeabilities were computed in a sequence of increasing and decreasing capillary pressures for drainage and waterflooding, respectively.

![Fig. 1: A two-dimensional section of X-ray microtomography scans of (a) Berea, (b) Clashach, and (c) Stainton samples. Voxel volume (resolution) is 5.789µm × 5.789µm × 5.789µm in (a) and (b), and 17.578µm × 17.578µm × 17.578µm in (c). The white horizontal bar in each image depicts 1.00mm.](image)

<table>
<thead>
<tr>
<th>Porosity</th>
<th>Berea</th>
<th>Clashach</th>
<th>Stainton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute permeability [m²]</td>
<td>7.8 × 10⁻¹³</td>
<td>8.1 × 10⁻¹⁵</td>
<td>1.8 × 10⁻¹³</td>
</tr>
<tr>
<td>Formation factor</td>
<td>16.0</td>
<td>27.2</td>
<td>55.3</td>
</tr>
</tbody>
</table>

Table 3: Petrophysical properties of the networks used in the pore-network simulations. Clay content was set to zero in all three networks.

Results

Capillary pressure. Brine saturation at the end of primary drainage is presented in Fig. 2 as a function of applied capillary pressure, $P_c$. Compared with the experiment are predictions by the pore network simulator (solid lines) and capillary pressure translated from mercury injection capillary pressure (MICP) measurements (Autopore IV 9520; Weatherford Laboratories, Stavanger, Norway) by the classic Leverett-J correlation (Bear 1972; dashed lines)

$$P_{cb}(S_w) = \frac{P_{cb}(S_w)}{\sigma_{\text{Hg}} \cos \theta_{\text{Hg}}} \frac{\cos \theta_{\text{Hg}}}{\cos \theta_{\text{Hg}}} = \frac{S_w}{\cos \theta_{\text{Hg}}}$$

where $S_w$ is the wetting phase saturation at the end of primary drainage, $\sigma$ is the interfacial tension, and $\theta$ is the macroscopic contact angle, with subscripts Hg and o/b denoting mercury/vacuum systems and oil/brine systems.
respectively. Here, $\sigma_{Hg} = 485\text{mN m}^{-1}$, $\theta_{Hg} = 130^\circ$, and $\theta_{o/b} = 0^\circ$ were assumed. The capillary pressure curve derived from MICP data agrees very well with ambient-condition measurements in Stainton (Fig. 2c). In contrast, the capillary pressure necessary to establish a given oil saturation during drainage at ETP was significantly larger than values predicted by Eq. 3 from MICP (Fig. 2a, b). This apparent discrepancy is attributed primarily to a premature termination of primary drainage in the ETP experiments (cf. Omorogie 1988). The magnitude of the discrepancy in brine saturations at a given $P_c$ is of the same order of magnitude or smaller than the measured $S_{nwr}$, which indicates that the effect on the initial-residual saturation curves is negligible except at the lowest $S_{nwr}$ (highest brine saturations). The consistent under-prediction of drainage capillary pressure by the pore network simulator, particularly at low brine saturations, has been reported in other studies (Cense and Marcelis 2008; Touati et al. 2009). This is attributed to two reasons. First, capillary pressure measurements do not necessarily record a position of true capillary equilibrium and hence, for a given saturation, may record a higher value than a quasi-static displacement. Second, the resolution of the images – from which the networks were extracted – is too coarse to resolve the fine details of the pore space (cf. Fig. 1). Furthermore, the predicted network permeabilities, Table 3, are consistently higher than the measured permeability, Table 2, indicating an over-estimation of the typical pore size. Indeed, the pore-throat radius distribution estimated from MICP measurements attributes 34.3%, 23.8%, and 99.5% of the pore volume in Berea, Clashach, and Stainton, respectively, to radii smaller than the corresponding voxel resolutions.

![Graph](image1.png)

**Fig. 2**: Brine saturation after primary drainage as a function of applied capillary pressure for (a) Berea, (b) Clashach, and (c) Stainton. Vertical bars in (a) and (b) depict the standard deviations over sampling time (4 to 54 hours). Horizontal bars depict the disagreement between the different methods of saturation determination for a given run. Where the bars are not visible, they are smaller than the size of the marker or are not available. Dashed lines are translated from MICP data using the Leverett-J correlation (Eq. 3), with oil/brine interfacial tension as summarized in Table 2 and $\theta_{o/b} = 0^\circ$. Solid lines depict predictions by Valvatne and Blunt (2004)’s network model.

**Capillary trapping.** Residual oil saturations are shown as a function of initial oil saturation in Figs. 3 and 4. Residual saturation increases monotonically, within experimental uncertainty, with increasing initial oil saturation for all samples. The increase is most rapid at low initial saturation and asymptotes at high initial saturation, clearly deviating from a linear dependence. Empirical correlations proposed by Land (1968; Eq. 1) and Spiteri et al. (2008; Eq. 2) are compared with experiment in Fig. 3. Here, $S_{nwr}^{\text{max}}$ and $S_w$ in Eq. 1 were taken as the mean residual and brine saturation of all runs for which the initial oil saturation is greater than 0.70; the
coefficients \( \alpha \) and \( \beta \) in Eq. 2 were determined by least-squares fitting to data. Both functions yield good agreement with the data.

The best-fit \((\alpha, \beta)\) for Berea and Clashach were \((\alpha, \beta) = (1.0, 0.5)\) (Table 4). This set of values was observed in Spiteri et al. (2008)’s simulations in a Berea sandstone network with an imposed macroscopic intrinsic contact angle of \( \theta = 70^\circ \) and randomly-assigned local contact angles in the range \( \theta \pm 20^\circ \). In contrast, the set of best-fit constants for Stainton, \((\alpha, \beta) = (0.69, 0.15)\), was not observed by Spiteri et al. (2008), which is not surprising, as this is a different rock sample.

![Graphs](a) (b) (c)

**Fig. 3:** Comparison of measured \( S_{\text{snwr}} \) \( (S_{\text{snwi}}) \) in (a) Berea, (b) Clashach, and (c) Stainton, with empirical expressions proposed by Land (1968; Eq. 1, dashed line) and Spiteri et al. (2008; Eq. 2, solid line). \( S_{\text{snwr}} \) and \( S_{\text{snwi}} \) in Eq. 1 were taken as the mean \( S_{\text{snwr}} \) and \( S_{\text{snwi}} \) of all runs for which initial oil saturation is greater than 0.70. Fitting parameters \( \alpha \) and \( \beta \) in Eq. 2 are listed in Table 4. Horizontal and vertical bars depict the disagreement between the different methods of saturation estimation for a given run. Where the bars are not visible, they are smaller than the size of the marker or are not available. The horizontal bar on the data point \((S_{\text{snwi}}, S_{\text{snwr}}) = (0.93, 0.47)\) in (c) extends to \( S_{\text{snwi}} = 1.04 \).

<table>
<thead>
<tr>
<th></th>
<th>Berea</th>
<th>Clashach</th>
<th>Stainton</th>
</tr>
</thead>
<tbody>
<tr>
<td>Land 1968</td>
<td>correlation coefficient</td>
<td>0.98</td>
<td>0.91</td>
</tr>
<tr>
<td>Spiteri et al. 2008</td>
<td>best-fit fitting parameters ( \alpha = 0.97; \beta = 0.49 )</td>
<td>( \alpha = 1.0; \beta = 0.50 )</td>
<td>( \alpha = 0.69; \beta = 0.15 )</td>
</tr>
<tr>
<td></td>
<td>correlation coefficient</td>
<td>0.98</td>
<td>0.92</td>
</tr>
</tbody>
</table>

**Table 4:** Correlation coefficient of data and empirical expressions proposed by Land (1968; Eq. 1) and Spiteri et al. (2008; Eq. 2), and fitting parameters \( \alpha \) and \( \beta \) as determined by least-squares fitting Eq. 2 to data.

Network model simulations agree well with experimental data in Berea and Clashach when the intrinsic contact angle is assigned uniformly within the range 35° to 65° (Fig. 4a and 4b, solid lines). These are effective local contact angles, accounting for grain roughness, converging/diverging pore geometry, and surface chemistry. This range coincides with the local contact angle distribution identified above by matching to Spiteri et al. (2008)’s simulations and with those used to predict relative permeability measured by Oak (1990) in Berea sandstone by Valvatne and Blunt (2004). In contrast, the network model over-predicts residual saturation in Stainton for both contact angle distributions considered (Fig. 4c). The poor agreement is attributed to the low resolution of the X-ray microtomography images from which the network was extracted (Fig. 1c).
Fig. 4: Comparison of measured (circle) and simulated $S_{\text{snw}}$ in (a) Berea, (b) Clashach, and (c) Stainton cores. Horizontal and vertical bars depict the disagreement between the different methods of saturation estimation for a given run. Where the bars are not visible, they are smaller than the size of the marker or are not available. The horizontal bar on the data point $(S_{\text{snwi}}, S_{\text{snwr}}) = (0.93, 0.47)$ in (c) extends to $S_{\text{snwi}} = 1.04$. Predictions by Valvatne and Blunt (2004)'s pore-network simulator assuming intrinsic contact angles distributed in the range 0° to 30° (dashed line) and 35° to 65° (solid line) are included.

Our laboratory measurements are compared with data reported in the literature in Fig. 5. Only experiments in two-phase, uniformly-wet systems in which the initial saturation was varied systematically and in which both primary drainage and secondary imbibition were achieved by displacement are included. Residual saturation was higher in the present experiments than previous data, which we suggest is related to the complexity of the pore geometry and the lower porosity of the sandstones we considered. Conversely, the lowest residual saturations occurred in porous media with the highest porosity and permeability, namely sand packs (green ×, +) and Kleppe et al. (1997)'s artificial core (grey −). Given this trend, it is interesting to note that the initial-residual saturation curves for Berea and Clashach coincide, despite the two cores having substantially different porosity and permeability (Table 2).
Fig. 5: Laboratory measurements of residual non-wetting phase saturation after wetting-phase flooding or spontaneous imbibition ($S_{nwr}$) as a function of initial non-wetting phase saturation after primary drainage ($S_{nwi}$) in Berea (red ●), in Stainton (blue ■), and in Clashach (yellow ▲) from the present study, in sand packs by Pentland et al. (2010) and Al Mansoori et al. (2010; air/brine), in five Dalton sandstone cores by Pickell et al. (1966), and in an artificial porous medium by Kleppe et al. (1997). For marker definition for data from the literature, see Table 1. Where uncertainty on data from literature is available, they are depicted by horizontal and vertical bars. Horizontal and vertical bars on data from the present study depict the disagreement between the different methods of saturation determination for a given run; where the bars are not visible, they are smaller than the size of the marker or are not available. The horizontal bar on the data point ($S_{nwi}$, $S_{nwr}$) = (0.93, 0.47) for Stainton extends to $S_{nwi}$ = 1.04.

The maximum residual is plotted as a function of porosity in Fig. 6a. Here, the maximum residual is taken as the mean of all runs for which initial oil saturation was greater than 0.70. For the three samples considered here, there is no clear dependence on porosity, contrary to observations of decreasing residual with increasing porosity reported by Jerauld (1997) and Holtz (2005). In contrast, the product of maximum residual saturation and porosity, i.e., the maximum residual as a fraction of the bulk volume, increases monotonically with porosity from 0.122 to 0.221 (Fig. 6b). In the context of carbon storage, this parameter, also referred to as the maximum bulk volume residual (e.g., Weiss et al. 2001), can be interpreted as the capillary trapping capacity of a potential storage site (Iglauer et al. 2009). The data are consistent with previous suggestions that the trapping capacity is largest in rocks with a porosity of around 0.2 (Iglauer et al. 2009).
Fig. 6: The maximum residual saturation (a) and the product of the maximum $S_{\text{wr}}$ and porosity (b), in Berea (*), Clashach (▲), and Stainton (■). Vertical bars in (a) depict the standard error of the mean.

Conclusions
Laboratory measurements of residual non-wetting phase saturation were presented as a function of initial saturation in three water-wet sandstones. The residual saturation increases monotonically, but with a decreasing gradient, with initial saturation. This trend is captured excellently by the trapping models proposed by Land (1968) and Spiteri et al. (2008). Network modeling on networks that were extracted directly from microtomography images of the pore space accurately predicted the residual saturations when a uniform distribution of intrinsic contact angle between 35° and 65° was assumed. The good agreement indicates that the effective contact angles in the systems we considered are not strongly wetting.

The residual saturations measured in the present study were higher than in previous displacement experiments, suggesting, for instance, that capillary trapping may be an effective way to store substantial quantities of CO$_2$ in aquifers. However, this presumes that super-critical CO$_2$/brine systems have similar wettabilities as in the present experiments; future work will focus on the capillary trapping of CO$_2$ at representative aquifer conditions.

Nomenclature
$P_c$ = applied capillary pressure, Pa
$S_{\text{ni}}$ = initial (post-primary drainage) non-wetting phase saturation
$S_{\text{wr}}$ = residual (post-imbibition) non-wetting phase saturation
$S_w$ = wetting phase saturation
$S_{\text{ic}}$ = connate (irreducible) wetting phase saturation
$\alpha$ = fitting parameter in Eq. 2
$\beta$ = fitting parameter in Eq. 2
$\sigma$ = interfacial tension, N m$^{-1}$
$\theta$ = macroscopic intrinsic contact angle, °
Acknowledgments
We would like to thank Shell under the Grand Challenge on Clean Fossil Fuels for supporting this work. The authors gratefully acknowledge Hu Dong for providing the pore networks used in this paper.

References


Holz, M.H. 2005. Reservoir characterization applying residual gas saturation modeling: example from the Starfak T1 reservoir, Middle Miocene Gulf of Mexico. Master’s thesis, University of Texas at Austin, Austin, TX.


Chapter 2

SPE 135261

Simulation of Flow and Dispersion on Pore-Space Images
Peyman Mostaghimi, SPE, Branko Bijeljic, SPE, and Martin J. Blunt, SPE, Imperial College London

This paper was prepared for presentation at the SPE Annual Technical Conference and Exhibition held in Florence, Italy, 19–22 September 2010.

Abstract
We simulate flow and transport directly on pore-space images obtained by micro-CT scanning of rock cores. An efficient Stokes solver is used to simulate low-Reynolds number flows. The flow simulator uses a finite-difference method along with a standard predictor-corrector procedure to decouple pressure and velocity. An algebraic multigrid technique solves the linear systems of equations. We then predict permeability and the results are compared with lattice Boltzmann numerical results and available experimental data.

For solute transport we apply a streamline-based algorithm that is similar to the Pollock algorithm common in field-scale reservoir simulation, but which employs a novel semi-analytic formulation near solid boundaries to capture, with sub-grid resolution, the variation in velocity near the grains. A random walk method accounts for molecular diffusion. The streamline-based algorithm is validated by comparison with published results for Taylor-Aris dispersion in a single capillary with a square cross-section. We then accurately predict available experimental data in the literature for longitudinal dispersion coefficient as a function of Peclet number. We introduce a characteristic length based on ratio of volume to pore/grain surface area that can be used for consolidated porous media to calculate Peclet number.

Introduction
Accurate modeling of solute dispersion at porous media is of great importance in many branches of science and engineering (Bear, 1988, Adler, 1992). To model the flow and transport at the pore scale to predict dispersion coefficients, the morphology of the pore space needs to be known. To obtain three-dimensional models of the pore space different methods can be applied. It is routine to obtain two-dimensional high-resolution images of rock (Zinszner and Meynot, 1982, Thovert et al., 1993) from which a three-dimensional representation is constructed. This can be achieved using object-based methods, where grains of different size and shape are deposited (Øren et al., 1998) or an image is obtained that reproduces the statistics of the two-dimensional pictures (Adler et al., 1990, Hazlett, 1997, Roberts, 1997, Okabe and Blunt, 2004). The advent of X-ray computed tomography has made it possible to obtain three-dimensional images with a resolution of a few microns (Coenen et al., 2004, Flannery et al., 1987) which is sufficient to capture the pore space of most sandstones.

The next step is modeling flow and transport through the pore space. Traditionally, network modeling has been applied to idealize the system as a lattice of wide pores connected by throats through which displacement and transport can be computed semi-analytically (Fatt, 1956, Blunt, 2001). Network models have been used widely to simulate dispersion at the pore scale (Bruderer and Bernabé, 2001, de Arcangelis et al., 1986, Sahimi et al., 1986, Sorbie and Clifford, 1991, Sahimi, 1995, Bijeljic et al., 2004, Bijeljic and Blunt, 2007, Acharya et al., 2007, Jha et al., 2008). However, a more direct approach is to simulate transport directly on a pore-space image that removes the need to extract an equivalent network with its inherent approximations (Maier et al., 2003, Coelho et al., 1997, Maier et al., 1998, Yao et al., 1997, Salles et al., 1993). Coelho et al. (1997) used a finite difference technique to solve for flow and simulated dispersion through packings of grains of arbitrary shape and found a good agreement with experimental results on unconsolidated bead packs and sandstones. Yao et al. (1997) used a similar method to solve for the flow field and modeled dispersion on a statistically reconstructed geometry of Vosges sandstone. The agreement of the model prediction with experimental measurements for dispersion coefficient was not satisfactory (Adler and Thovert, 1998). Maier et al. (1998) used the lattice Boltzmann technique to model dispersion in a three-dimensional model of a sphere packing and successfully compared their numerical results with NMR experiments for a wide range of dimensionless Peclet number. Lattice Boltzmann simulation has also been used the dispersion studies in cylinder packs, sphere packs, and uniform porous media by Zhang and Lv (2007), Maier et al. (2007), Hlushkou (2007) and Khirevich et al. (2010). Garmeh et al. (2007) used a finite-element scheme to solve for fluid flow and once the velocities were
calculated, they solved the convection-diffusion equation to model dispersion in two and three dimensional packings of spheres.

These previous studies have been able to reproduce the rich behavior of the dispersion coefficient as a function of Peclet number, demonstrating how the subtle interplay of advection and diffusion in a disordered pore space affects transport. However, the models are restricted to reconstructed or simple media or network models that may miss some details of the pore structure. These limitations may be important when exploring possible extensions of the work to embrace reactive transport or multiphase flow.

We simulate the flow and transport directly on pore-space images obtained from micro-CT scanning by solving the Stokes equations in the pore space and by applying a novel streamline-based algorithm which employs a new semi-analytic formulation near solid boundaries to capture, with sub-grid resolution, the variation in velocity near the solid. A random walk method accounts for diffusion. We successfully predict different dispersion regimes in laminar flow and our results show a very good agreement with previous numerical and experimental studies. This is a first step towards the simulation of reactive transport, multiphase flow and transport in complex media, such as carbonates.

Method

Our algorithm to model the dispersion of non-reactive particles is

1. Solve for Stokes flow in the pore space.
2. Inject particles.
3. In each time step:
   3.1. Move particles by advection using streamline tracing.
   3.2. Move particles by diffusion using the random-walk method.
4. Compute dispersion coefficients from the overall movement of an ensemble of particles.

For step 1, we consider each void voxel in an image obtained by micro-CT scanning as a grid block in our algorithm for simulating fluid flow. The Stokes equations combined with the continuity equation are used to solve for incompressible steady viscous flow.

\[ \nabla \cdot \mathbf{v} = 0 \]  
\[ \mu \nabla^2 \mathbf{v} = \nabla P \]  
\[ \mathbf{v} = 0 \] on grains

where \( \mathbf{v} = (u, v, w) \) is the vector of velocity, \( P \) is pressure, and \( \mu \) is the fluid viscosity.

For discretizing the governing equations in three dimensions, we have applied the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE), pioneered by Patankar and Spalding (1972) that has been widely applied to the fields of computational fluid dynamics and numerical heat transfer. This method converts the continuity equation into a numerical equation to correct for pressure, resulting in an iterative solution. We use structured marker-and-cell (MAC) gridding (Peyret and Taylor, 1985, Harlow and Welch, 1965). MAC gridding means that the nodes where different variables are being estimated are not the same: pressures are determined at the center of cells and normal velocities are defined at cell faces. Because of the MAC gridding system, the velocity component parallel to a solid surface is not automatically located at the surface of solid voxels. We have applied a non-centered higher-order finite-difference scheme to rewrite the formulation for cells in the neighborhood of solid voxels. In this technique, we put the no-slip condition exactly on the solid surface. To solve the resultant linear systems of equations, an algebraic multigrid technique has been used (Stüben, 2001, Hackbusch, 1985).

In step 2, 3,000 particles are injected uniformly in the pore space of the inlet face. We also performed some computations with 10,000 particles. The difference in predicted dispersion coefficient was typically 3% and always less than 10%.

In step 3, we move particles in each time step due to advection and diffusion. For advection, step 3.1, particles are moved along streamlines that follow the velocity field. Pollock (1988) suggested a linear interpolation between the normal velocities on opposite faces of a grid block. This is consistent with a divergence-free velocity field and the finite difference approximation used to compute pressure and flow, as well as ensuring continuity of normal velocity. This method is now standard in field-scale streamline-based reservoir simulation (Batycky et al., 1997). For voxels with no solid boundaries we use the same method here, which is – again – consistent with incompressible flow and the finite difference approximations used. However, at a solid surface we need to impose a zero normal and tangential velocity, which is not consistent with the Pollock algorithm that would only ensure that the velocity normal to the solid is zero. To overcome this limitation and to impose a strictly vanishing velocity at the solid we have introduced a novel semi-analytic formulation for the streamlines in voxels containing a solid boundary. The details of the formulation – given in the Appendix – vary with the location and numbers of solid faces in a voxel, and give a tangential velocity that varies quadratically.
away from a boundary. The result is a method that strictly obeys the boundary conditions and allows a semi-analytic tracing of streamlines throughout the domain.

We can calculate the time needed for a particle inside the voxel to exit, $\Delta \tau$. The time step is $\Delta t$. If $\Delta \tau > \Delta t$ the particle remains within the voxel, whereas for $\Delta \tau < \Delta t$ the particle enters a new block. We reduce the time step by $\Delta \tau$ and find the $\Delta \tau$ for the new grid block and repeat the previous procedure.

In step 3.2, we model movement of particles due to diffusion using a random walk method which is an established Lagrangian technique to model conservative and reactive transport in porous media (Ahlstrom et al., 1977, Tompson, 1993, Maier et al., 1998). A particle instantaneously jumps over a fixed distance $\zeta$ in a random direction:

$$\zeta = \sqrt{6D_m \Delta t}$$  (4)

The coordinates of the particle after diffusion is:

$$x = x_p + \zeta \sin \phi \cos \theta$$  (5)

$$y = y_p + \zeta \sin \phi \sin \theta$$  (6)

$$z = z_p + \zeta \cos \phi$$  (7)

$\theta$ is a random number in the range of $(0, 2\pi)$ and $\phi$ is another random number in the range of $(0, \pi)$.

We use a reflecting boundary condition if the random jump places a particle in the solid. We allow multiple reflections if the particle encounters more than one solid surface.

If a particle exits the system during its advection motion, it will be re-injected at the inlet face at a flow-weighted random location; if the particle leaves during the diffusive step, it will be re-injected by an area-weighted random position on the inlet face. In addition, if due to diffusion, a particle exits through the inlet, it will be injected on the outlet face, again assigning a random area-weighted position.

In step 4, the longitudinal dispersion coefficient is estimated by calculating the variance of the distance traveled by particles in the main direction of flow:

$$D_L = \frac{1}{2} \frac{d\sigma^2}{dt}$$  (8)

where $\sigma$ is the variance of the particle displacement:

$$\sigma = \langle x(t) - < x(t) > \rangle^2$$  (9)

The time step for the simulation is $10^{-4}$ s. This value has been chosen to restrict the motion of particle at each time step be less than one voxel. The molecular diffusion coefficient in our simulations is $10^{-9}$ m$^2$s$^{-1}$. Therefore at each time step the size of the diffusive step, Eq. (4), is 0.77$\mu$m which is less than the size of each voxel (5.345 $\mu$m). To study the effect of Peclet number, we vary the flow rate. We run the simulations until we reach an asymptotic regime where $D_L$ computed by Eq. (8) no longer changes with time.

Validation

We first considered a very simple case: flow between two parallel plates. Fig. 1 shows a comparison of the computed velocity for different numbers of grid blocks across the opening compared to the analytical solution (White, 1999). For two or more cells between the plates, the face velocities are found to within machine error. We see the same for flow through a cylinder of square cross-section. This result suggests that we can obtain a reasonable representation of the flow field even when the flow channels are represented with only a few voxels in the micro-CT image.

Furthermore, we modeled three-dimensional flow around a cubic obstacle. The model contains 50×50×50 voxels and the cubic obstacle is located at the center of the opening with dimension of 15×40×40 voxels. We compared our results against the free open source CFD toolbox, OpenFOAM (www.openfoam.com). The difference in average estimated velocity is less than 0.1% between the two codes and the local velocities differ by no more than 0.5%. In this, and all subsequent simulations, at each iteration, we calculate the volumetric flow rate passing through each layer of the system normal to main direction of flow. We iterate for velocity and pressure, until the difference between flows of each pair of layers normalized by average flow rate of whole system is less than five percent.
We then modeled single-phase flow on a series of sand packs and sandstone micro-CT images. All the images are 300×300×300 voxels. The predicted permeabilities are in good agreement with published lattice-Boltzmann results (Dong and Blunt, 2009), see Table 1.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Resolution (μm)</th>
<th>Porosity (%)</th>
<th>LBM permeability (D)</th>
<th>Predicted permeability (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandpack LV60A</td>
<td>10.0</td>
<td>38</td>
<td>35</td>
<td>39</td>
</tr>
<tr>
<td>Sandpack LV60B</td>
<td>8.6</td>
<td>37</td>
<td>31</td>
<td>34</td>
</tr>
<tr>
<td>Sandpack LV60C</td>
<td>10.0</td>
<td>37</td>
<td>19</td>
<td>22</td>
</tr>
<tr>
<td>Sandpack F42A</td>
<td>10.0</td>
<td>33</td>
<td>31</td>
<td>59</td>
</tr>
<tr>
<td>Sandpack F42B</td>
<td>10.0</td>
<td>33</td>
<td>52</td>
<td>62</td>
</tr>
<tr>
<td>Sandpack F42C</td>
<td>10.0</td>
<td>33</td>
<td>50</td>
<td>47</td>
</tr>
<tr>
<td>Sandstone 1</td>
<td>8.7</td>
<td>20</td>
<td>1.7</td>
<td>1.9</td>
</tr>
<tr>
<td>Sandstone 2</td>
<td>5.0</td>
<td>25</td>
<td>3.9</td>
<td>2.9</td>
</tr>
<tr>
<td>Sandstone 3</td>
<td>9.1</td>
<td>17</td>
<td>0.22</td>
<td>0.19</td>
</tr>
<tr>
<td>Sandstone 4</td>
<td>9.0</td>
<td>17</td>
<td>0.26</td>
<td>0.31</td>
</tr>
<tr>
<td>Berea Sandstone</td>
<td>5.3</td>
<td>20</td>
<td>1.3</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Experimental measurements of permeability, measured on larger core samples, are also available in some cases (Talabi and Blunt, 2010). The LV60 and F42 sand packs have permeabilities of 32.2 ± 0.3 D and 42 ± 4 D respectively which are in reasonably good agreement with the average of our numerical predictions on three different images. The core-measured permeability of Berea is 0.7 D, which is half the computed value. This is likely to be due to heterogeneity and the large size difference between the micro-CT sample that is only 1.5 mm across and the core sample with a length of 7 cm. The Berea image was used for subsequent analysis to compute dispersion coefficient.

To validate our particle tracking algorithm, we simulated dispersion in a single capillary with square cross section and compared it with the previous study by Bruderer and Bernabé (2001) and Bijeljic et al. (2004). The square opening was 50×50 grid blocks across. The characteristic length used in calculating the Peclet number is the half-length of the side of the square. Aris (1956) showed that the longitudinal dispersion coefficient in a single capillary is proportional to the square of the Peclet number:

\[
\frac{D_L}{D_m} = \kappa P_e^2
\]

where \( \kappa \) is a constant coefficient that depends on the shape of the capillary. Bruderer and Bernabé (2001) found that for capillary with square cross section \( \frac{1}{\kappa} = 30.4 \pm 2.6 \) and Bijeljic et al (2004) obtained \( \frac{1}{\kappa} = 29.24 \).

The best fit to our results shows \( \frac{1}{\kappa} = 28.93 \) which is in good agreement with previous studies.
Results
Flow was computed on the Berea sandstone image. Fig. 2 shows the geometry, pressure distribution and velocity in the model.

Figure 2. The geometry – gray shows the pore volume with scale bar in millimeter unit (a), normalized pressure field with a unit pressure difference across the model (b), normalized flow field (c) for the Berea sandstone image.

Once the velocity field was obtained, we have solved for transport over a range of Peclet number, defined as:

\[ Pe = \frac{u_{avg} L}{D_m} \]

where \( u_{avg} \) is the average velocity and \( L \) is the characteristic length. Average velocity is calculated from:

\[ u_{avg} = \frac{Q}{A \phi} \]
where $Q$ is volumetric flow rate in the whole model, $\phi$ is porosity, and $A$ is the area perpendicular to the main direction of flow (the area of the inlet face).

For experiments on unconsolidated porous media, the average diameter of grains can be used as the characteristic length; however, this is not available directly from a micro-CT image of a consolidated rock. We defined the characteristic length based on a cubic packing of regular spheres. For this idealized system, the grain diameter is $\pi V/S$, where $V$ is the volume of the porous medium (pore plus grain) and $S$ is the area of the pore/grain interface. We use the same definition for our images, since the volume and pore/grain area are readily computed:

$$L = \frac{\pi V}{S}$$

(13)

We obtained $L = 131.13 \, \mu m$ for Berea.

Fig. 3 shows our numerical results for longitudinal dispersion coefficient as function of Peclet number along with published experimental data on unconsolidated bead packs obtained from breakthrough curves (Pfannkuch, 1963), data on packs of plastic beads obtained by Pulsed-field-gradient NMR (Ding and Candela, 1996), data on bead packs obtained by magnetic resonance imaging (Seymour and Callaghan, 1997, Kandhai et al., 2002, Khrapitchev and Callaghan, 2003), data on a packed column of activated carbon grains with a bimodal pore size distribution based on radioactive dispersion (Drazer et al., 1999), data bead packs obtained by planar laser-induced fluorescence method (Stöhr, 2003), data on sandy soil column measured by electrical resistance between rod electrodes and fitting analytical models of advection-dispersion equation to concentration breakthrough curves at axial positions along the column (Aggelopoulos and Tsakiroglou, 2007), and data on a planar glass-etched pore network obtained using an experimental set-up detecting color changes caused during the mixing (Theodoropoulou, 2007).
Figure 3. Reduced dispersion coefficient (ratio of longitudinal dispersion to molecular diffusion coefficient) against dimensionless Peclet number. Different characteristic lengths have been used for estimation of Peclet number in the numerical and experimental results. The vertical bars show results for sandstone samples in the restricted diffusion regime at low Peclet number (Dullien, 1992, Frosch et al., 2000).

The agreement between experiment and the numerical results is good, although there is a tendency for the predicted curve to be shifted to slightly larger Pe, indicating, perhaps, the difference in definition of characteristic length, particularly at high Pe. The reduced dispersion coefficient is less than unity in the low Peclet regime because of the presence of the solid, restricting diffusion: our numerical results here correspond to measurements on sandstones of similar porosity, but under-estimate the dispersion for bead packs. At higher Pe we see a transition regime followed by an approximate power-law regime with $D_L \sim Pe^\delta$ with an exponent $\delta = 1.2$, consistent with measurements and other analyses (Brigham et al., 1961, Salter and Mohanty, 1982, Kinzel and Hill, 1989, Bijeljic et al., 2004).

The mechanical dispersion regime can be observed for Pe > 600 where $D_L \sim Pe$. In this regime, the effects of molecular diffusion can be ignored and transport is dominated by advection only. We underestimate the dispersion coefficient in this regime. This could be due to inertial and turbulent effects which have been ignored in our flow simulation (Sahimi, 1995).

Discussion and Conclusions
We have described a method to simulate flow and transport in pore-space images. A finite-difference method is used to compute Stokes flow: we show that for pores of constant cross-section, the velocity is obtained to machine accuracy if there are two or more voxels across the opening. This is important for practical application, where there is a trade-off between image resolution and the size of the sample, meaning that many pores are only resolved with a few voxels. We compute the permeability for a series of images and good agreement is obtained with experimental data and predictions using the lattice-Boltzmann technique.
We have developed a new streamline-tracing technique to model advection. For voxels with no solid boundary, the method is identical to the Pollock (1988) algorithm that is now standard in field-scale streamline-based reservoir simulators. Where a solid boundary is present, zero tangential and normal velocity is strictly imposed, resulting in a quadratic variation of tangential velocity away from the boundary. A semi-analytic description of the velocity field within a grid block is obtained for all combinations of solid boundaries. An ensemble of particles is moved along streamlines, while a random walk method is used to capture molecular diffusion.

We predict asymptotic longitudinal diffusion coefficient as a function of Peclet number. We introduce a new characteristic length based on the ratio of volume to pore/grain surface area, applicable to images where a typical grain size cannot be easily obtained. Our results compare well with measurements in the literature and show the different regimes, including restricted dispersion, a power-law dependence of dispersion coefficient on Peclet number and the mechanical dispersion regime at high Peclet number.

This work serves as a basis for further analysis of dispersion on a variety of different porous media. We plan to predict longitudinal and transverse dispersion for different rock types, including pre-asymptotic behavior where the apparent dispersivity evolves over time. Our methodology is also a useful first step in the development of a rigorous pore-scale treatment of geochemical reactions.

**Appendix – Streamline Tracing Algorithm**

We use the method of (Pollock, 1988) in voxels with no solid boundaries. The velocity normal to a face varies linearly across the block – see Fig. 4.

\[
\begin{align*}
    u &= \frac{u_2 - u_1}{\Delta x} (x - x_i) + u_i \\
    v &= \frac{v_2 - v_1}{\Delta y} (y - y_i) + v_i \\
    w &= \frac{w_2 - w_1}{\Delta z} (z - z_i) + w_i
\end{align*}
\]

where \( u, v \) and \( w \) are the components of velocity in the \( x, y \) and \( z \) directions respectively. \( u_i \) and \( u_2 \) are the computed velocities on the faces normal to the \( x \) direction, \( v_i \) and \( v_2 \) are velocities on the faces normal to the \( y \) direction, while \( w_i \) and \( w_2 \) are the velocities on the faces normal to the \( z \) direction. \( \Delta x, \Delta y \) and \( \Delta z \) are the dimensions of the cell in the respective coordinate directions – see Fig. 4. The time necessary for a particle to exit the \( x \)-face is \( \Delta \tau_x \), and the time to exit \( y \)-face and \( z \)-face are \( \Delta \tau_y \) and \( \Delta \tau_z \) respectively.

\[
\begin{align*}
    \Delta \tau_x &= \frac{\Delta x}{u_2 - u_1} \ln \frac{u_2 \Delta x}{u_i \Delta x + (u_2 - u_i)(x_p - x_i)} \\
    \Delta \tau_y &= \frac{\Delta y}{v_2 - v_1} \ln \frac{v_2 \Delta y}{v_i \Delta y + (v_2 - v_i)(y_p - y_i)} \\
    \Delta \tau_z &= \frac{\Delta z}{w_2 - w_1} \ln \frac{w_2 \Delta z}{w_i \Delta z + (w_2 - w_i)(z_p - z_i)}
\end{align*}
\]

where \((x_p, y_p, z_p)\) are the initial coordinates of the particle. The particle leaves the grid block with the smallest value among the computed increments in time of flight.

\[
\Delta \tau = \min(\Delta \tau_x, \Delta \tau_y, \Delta \tau_z)
\]

And the exit location can be obtained from equations below:

\[
\begin{align*}
    x_e &= x_i - \frac{u_1 \Delta x}{u_2 - u_1} + \left[ \frac{u_1 \Delta x}{u_2 - u_i} + (x_p - x_i) \right] e^{\frac{u_1 - u_i}{\Delta x} \Delta \tau_x} \\
    y_e &= y_i - \frac{v_1 \Delta y}{v_2 - v_1} + \left[ \frac{v_1 \Delta y}{v_2 - v_i} + (y_p - y_i) \right] e^{\frac{v_1 - v_i}{\Delta y} \Delta \tau_y} \\
    z_e &= z_i - \frac{w_1 \Delta z}{w_2 - w_1} + \left[ \frac{w_1 \Delta z}{w_2 - w_i} + (z_p - z_i) \right] e^{\frac{w_1 - w_i}{\Delta z} \Delta \tau_z}
\end{align*}
\]
Figure 4. Velocity interpolation if there is no bounding solid voxel. This is the Pollock (1988) algorithm, standard in field-scale streamline-based reservoir simulation.

This method is used if no face is solid. In other cases, we categorize different cases where one or more nearest neighbor voxel is solid. In all cases the velocity field everywhere is divergence free and obeys a zero velocity at all solid boundaries.

1. One of the neighboring voxels is solid

This condition can happen in six different ways depending on which face is solid. We show the formulation for one of the cases; the formulation for the others will be similar. If the solid voxel blocks the $x_2$-face (Fig. 5), the velocity distribution is:

$$u = \frac{u_1}{\Delta x^2}(x_2 - x)^2$$ \quad (24)

$$v = \frac{2v_1}{\Delta x}(x_2 - x) + \frac{2(v_2 - v_1)}{\Delta x \Delta y}(x_2 - x)(y - y_1)$$ \quad (25)

$$w = \frac{2w_1}{\Delta x}(x_2 - x) + \frac{2(w_2 - w_1)}{\Delta x \Delta z}(x_2 - x)(z - z_1)$$ \quad (26)

The time of flight increment across the block is:

$$\Delta \tau_x = \frac{\Delta x^2}{u_1} \left( \frac{1}{\Delta x} - \frac{1}{x_2 - x_p} \right)$$ \quad (27)

$$\Delta \tau_y = \frac{\Delta x^2}{u_1(x_2 - x_p)} \left( \frac{v_2 \Delta y}{v_1 \Delta y + (v_2 - v_1)(y_p - y_1)} \right) \frac{u_1 \Delta y}{2 \Delta x (v_2 - v_1)} - \frac{\Delta x^2}{u_1(x_2 - x_p)}$$ \quad (28)

$$\Delta \tau_z = \frac{\Delta x^2}{u_1(x_2 - x_p)} \left( \frac{w_2 \Delta z}{w_1 \Delta z + (w_2 - w_1)(z_p - z_1)} \right) \frac{u_1 \Delta z}{2 \Delta x (w_2 - w_1)} - \frac{\Delta x^2}{u_1(x_2 - x_p)}$$ \quad (29)

$$\Delta \tau = \min(\Delta \tau_x, \Delta \tau_y, \Delta \tau_z)$$ \quad (30)

And the exit location is:
\[ x_{e} = x_2 - \left( \frac{\Delta \tau}{\Delta t^2} u_1 + \frac{1}{x_2 - x_p} \right)^{-1} \]  

(31)

\[ y_{e} = y_1 - \frac{v_1 \Delta y}{v_2 - v_1} + \frac{v_1 \Delta y + (v_2 - v_1)(y_p - y_1)}{(v_2 - v_1) \Delta y} \left( 1 + \frac{u_1(x_2 - x_p)}{\Delta t^2 \Delta \tau} \right) \frac{2 \Delta \tau (v_2 - y_1)}{u_2 \Delta y} \]  

(32)

\[ z_{e} = z_1 - \frac{w_1 \Delta z}{w_2 - w_1} + \frac{w_1 \Delta z + (w_2 - w_1)(z_p - z_1)}{(w_2 - w_1) \Delta z} \left( 1 + \frac{u_1(x_2 - x_p)}{\Delta t^2 \Delta \tau} \right) \frac{2 \Delta \tau (w_2 - w_1)}{u_1 \Delta z} \]  

(33)

Figure 5. There is one solid nearest-neighbor voxel.

2. Two of the neighboring voxels are solid

This condition can happen for fifteen cases that can be categorized in two groups: (1) when the two solid voxels block opposing faces (this can happen in three cases, for the x, y- or z-directions); and (2) when the solid voxels are located on adjoining faces that can happen in twelve cases.

2.1. There are two opposing solid faces

We show the equations where both solid blocks normal to the x-direction (Fig. 6); formulations for the two other cases can be obtained in the same way. Since both x-faces are blocked, the velocity component in the x direction is zero. The velocity distribution is:

\[ u = 0 \]  

(34)

\[ v = \frac{6v_1}{\Delta x^2}(x_2 - x)(x - x_1) + \frac{6(v_2 - v_1)}{\Delta x^2 \Delta y}(x_2 - x)(x - x_1)(y - y_1) \]  

(35)

\[ w = \frac{6w_1}{\Delta x^2}(x_2 - x)(x - x_1) + \frac{6(w_2 - w_1)}{\Delta x^2 \Delta z}(x_2 - x)(x - x_1)(z - z_1) \]  

(36)

The time of flight increment across the block is:

\[ \Delta \tau_y = \frac{\Delta x^2 \Delta y}{6(v_2 - v_1)(x_2 - x_p)(x_p - x_1)} \ln \left( \frac{v_2 \Delta y}{v_1 \Delta y + (v_2 - v_1)(y - y_1)} \right) \]  

(37)

\[ \Delta \tau_z = \frac{\Delta x^2 \Delta z}{6(w_2 - w_1)(x_2 - x_p)(x_p - x_1)} \ln \left( \frac{w_2 \Delta z}{w_1 \Delta z + (w_2 - w_1)(z - z_1)} \right) \]  

(38)

\[ \Delta \tau = \min(\Delta \tau_y, \Delta \tau_z) \]  

(39)

Having the time of flight, the exit location is:
\[ x_e = x_p \]  
\[ y_e = y_1 - \frac{v_1 \Delta y + (v_2 - v_1)(y_p - y_1)}{(v_2 - v_1) \Delta y} e^{\frac{6(v_2-v_1)(x_p-x_1)(x_2-x_p)}{\Delta x^3} \Delta t} \]  
\[ z_e = z_1 - \frac{w_1 \Delta z + (w_2 - w_1)(z_p - z_1)}{(w_2 - w_1) \Delta z} e^{\frac{6(w_2-w_1)(x_1-x_p)(x_2-x_2-x_p)}{\Delta z^3} \Delta t} \]

Figure 6. Two opposing voxels are solid.

Figure 7. Two adjacent voxels are solid.

2.2. Two adjacent voxels are solid

We show the equations for the case where one of solid voxels blocks the \( x_2 \)-face and the other blocks the \( z_1 \)-face (Fig. 7); for the other eleven cases, the formulation can be obtained similarly. The velocity distribution is:
The time of flight increment across the block is:

$$\Delta \tau_x = \frac{\left(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p\right)0.5 \Delta x^2 \Delta y}{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5} \left[1 - \frac{\Delta x}{x_p - x_l}\right]}$$

$$\Delta \tau_y = \frac{\left(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p\right)0.5 \Delta y^2 \Delta x}{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5} \left[1 - \frac{\Delta y}{y_p - y_l}\right]}$$

$$\Delta \tau_z = \frac{\Delta x^2 \Delta y^2}{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5} \left[\left(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p\right)0.5 - (\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} - \Delta z)^{0.5}\right]}$$

$$\Delta \tau = \min(\Delta \tau_x, \Delta \tau_y, \Delta \tau_z)$$

And the exit location coordinates are:

$$x_e = x_1 + (x_p - x_0)(1 - \left[\frac{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5}}{(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p)^{0.5} \Delta y^2 \Delta x} - \frac{u_2 \Delta y + v_2 \Delta x}{u_2 \Delta y + v_2 \Delta x}\right])$$

$$y_e = y_1 + (y_p - y_0)(1 - \left[\frac{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5}}{(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p)^{0.5} \Delta x^2 \Delta y} - \frac{u_2 \Delta y + v_2 \Delta x}{u_2 \Delta y + v_2 \Delta x}\right])$$

$$z_e = z_1 + \frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x}\left[\left(\frac{w_1 \Delta x \Delta y}{u_2 \Delta y + v_2 \Delta x} + z_1 - z_p\right)0.5 - \frac{2(x_p - x_l)(y_p - y_l)[w_1 \Delta x \Delta y(u_2 \Delta y + v_2 \Delta x) + (z_1 - z_p)(u_2 \Delta y + v_2 \Delta x)^2]^{0.5}}{(\Delta x^2 \Delta y^2) - \Delta \tau}^2\right]$$
3. Three of the neighboring voxels are solid

This condition can happen in twenty ways depending on the locations of solid voxels and can be categorized in two groups. The three surrounding solid voxels block all three coordinate directions; one solid voxel is on x-, the other one on y- and the last one on the z-direction face. This condition can happen in eight different ways. The second group is when two of the three solid voxels block the same direction (for example both are in the x-direction) and the third one is in another direction.

3.1. Three of the neighboring voxels are solid that block in different directions

We have one solid voxel in each coordinate direction and in each direction we have two potential positions, so there are eight cases. We show the equations for one of them (Fig. 8) and the others can be calculated in the same way.

\[ u = \frac{4u_1}{\Delta x^2 \Delta y \Delta z} (x_2 - x)^2 (y_2 - y)(z - z_1) \]  
(53)

\[ v = \frac{4v_1}{\Delta x \Delta y^2 \Delta z} (x_2 - x)(y_2 - y)^2 (z - z_1) \]  
(54)

\[ w = \frac{4w_2}{\Delta x \Delta y \Delta z^2} (x_2 - x)(y_2 - y)(z - z_1)^2 \]  
(55)

The time of flight increment across the block can be calculated:

\[ \Delta \tau_x = \frac{\Delta x^2 \Delta y \Delta z}{4(x_2 - x_p)(y_2 - y_p)(z - z_1)u_1} \ln \left( \frac{x_2 - x_p}{\Delta x} \right) \]  
(56)

\[ \Delta \tau_y = \frac{\Delta x \Delta y^2 \Delta z}{4(x_2 - x_p)(y_2 - y_p)(z - z_1)v_1} \ln \left( \frac{y_2 - y_p}{\Delta y} \right) \]  
(57)

\[ \Delta \tau_z = \frac{\Delta x \Delta y \Delta z^2}{4(x_2 - x_p)(y_2 - y_p)(z - z_1)w_2} \ln \left( \frac{z - z_1}{\Delta z} \right) \]  
(58)

\[ \Delta \tau = \min(\Delta \tau_x, \Delta \tau_y, \Delta \tau_z) \]  
(59)

And the exit location is:

\[ x_e = x_2 - (x_2 - x_p)e^{-\frac{4u_1(x_2 - x_p)(y_2 - y_p)(z - z_1)}{\Delta x \Delta y \Delta z} \Delta \tau} \]  
(60)

\[ y_e = y_2 - (y_2 - y_p)e^{-\frac{4v_1(x_2 - x_p)(y_2 - y_p)(z - z_1)}{\Delta x \Delta y \Delta z} \Delta \tau} \]  
(61)

\[ z_e = z_1 + (z - z_1)e^{\frac{4w_2(x_2 - x_p)(y_2 - y_p)(z - z_1)}{\Delta x \Delta y \Delta z} \Delta \tau} \]  
(62)
3.2. Three of the neighboring voxels are solid and two of them are in the same direction

This condition can happen for twelve cases; we show the equations for the case where two solid voxels have blocked the x-direction (Fig. 9). Therefore, the velocity component in the x-direction will be zero. The velocity distribution is:

\[ u = 0 \]  

\[ v = \frac{12v_1}{\Delta x^2 \Delta z} (x_2 - x)(x - x_1)(z - z_1) + \frac{12(v_2 - v_1)}{\Delta x^2 \Delta y \Delta z} (x_2 - x)(x - x_1)(z - z_1)(y - y_1) \]  

\[ w = \frac{6w_2}{\Delta x^2 \Delta z^2} (x_2 - x)(x - x_1)(z - z_1)^2 \]  

The time of flight increment across the block is:

\[ \Delta \tau_y = \frac{\Delta x^2 \Delta z^2}{6w_2(x_2 - x_p)(x_p - x_1)(z_p - z_1)} (1 - \frac{v_1 \Delta y + (v_2 - v_1) \Delta y}{v_1 \Delta y + (v_2 - v_1)(y_p - y_1)} \frac{-w_2 \Delta y}{2(v_2 - v_1) \Delta z}) \]
\[
\Delta \tau_z = \frac{\Delta x^2 \Delta z^2}{6(x_2 - x_p)(x_p - x_1)w_2} \left( \frac{1}{z_p - z_1} - \frac{1}{\Delta z} \right) \\
\Delta \tau = \min(\Delta \tau_x, \Delta \tau_z) 
\]

And the exit coordinates are:

\[
x_e = x_p \\
y_e = y_1 - \frac{v_1 \Delta y}{v_2 - v_1} + \frac{6w_2(x_2 - x_p)(x_p - x_1)(z_p - z_1)}{\Delta x^2 \Delta z^2} \Delta \tau \\
z_e = z_1 + \frac{\Delta x^2 \Delta z^2(z_p - z_1)}{\Delta x^2 \Delta z^2 - 6w_2(x_2 - x_p)(x_p - x_1)(z_p - z_1)\Delta \tau} 
\]

4. Four of the neighboring voxels are solid

This condition can happen in fifteen cases that can be categorized in two groups: (1) when in two directions we have just one solid voxel and in the third direction there are two solid voxels; and (2) when there is no solid voxel in one of the three coordinate directions.

4.1. Four of the neighboring voxels are solid and three of them are in three different directions

We consider the case where there are two solid voxels in the x-direction and the other two are in the y and z directions (Fig. 10). Since the x-direction is blocked from both sides, the velocity component in the x direction is zero. The equations for the other cases can be obtained similarly.

\[
u = 0 \\
v = \frac{12v_1}{\Delta x^2 \Delta y^2 \Delta z} (x_2 - x)(x - x_1)(y_2 - y)^2(z - z_1) \\
w = \frac{12w_2}{\Delta x^2 \Delta y \Delta z^2} (x_2 - x)(x - x_1)(y_2 - y)(z - z_1)^2 
\]

The time of flight increment is:

\[
\Delta \tau_y = \frac{\Delta x^2 \Delta y^2 \Delta z}{12v_1(y_2 - y_p)(z_p - z_1)(x_2 - x_p)(x_p - x_1)} \ln\left(\frac{y_2 - y}{y_2 - y_p}\right) \\
\Delta \tau_z = \frac{\Delta x^2 \Delta z^2 \Delta y}{12w_1(y_2 - y_p)(z_p - z_1)(x_2 - x_p)(x_p - x_1)} \ln\left(\frac{z_p - z}{z_p - z_1}\right) \\
\Delta \tau = \min(\Delta \tau_y, \Delta \tau_z) 
\]

And the exit location is:

\[
x_e = x_p \\
y_e = y_2 - (y_2 - y_p) e^{-12v_1(x_2 - x_p)(x_p - x_1)(y_2 - y_p)(z_p - z_1)\Delta \tau \Delta z^2 \Delta y} \\
z_e = z_1 + (z_p - z_1) e^{\frac{12w_1(x_2 - x_p)(x_p - x_1)(y_2 - y_p)(z_p - z_1)}{\Delta x^2 \Delta z^2 \Delta y} \Delta \tau} 
\]
4.2. **Four of the neighboring voxels are solid and they block two directions**

We present the equations for the case where there is no solid voxel in the $z$ direction and the velocity components in $x$ and $y$ directions are zero (Fig. 11). The formulation for the other cases can be obtained similarly.

\[
\begin{align*}
    u &= 0 \\
    v &= 0 \\
    w &= \frac{36w_1}{\Delta x^2 \Delta y^2} (x_2 - x)(x - x_1)(y_2 - y)(y - y_1)
\end{align*}
\]

The time of flight increment across the block is:

\[
\Delta \tau = \frac{\Delta x^2 \Delta y^2}{36w_1 (x_2 - x_p)(x_p - x_1)(y_2 - y_p)(x_p - x_1)}
\]

The exit location is:

\[
\begin{align*}
    x_e &= x_p \\
    y_e &= y_p
\end{align*}
\]
\[ z_e = z_p + \frac{36 \rho}{\Delta x \Delta y} (x_2 - x)(x - x_1)(y_2 - y)(y - y_1) \Delta \tau \] (87)

If five or six faces are solid, the velocity everywhere is zero.

**Acknowledgments**

We thank the sponsors of the Imperial College Consortium on Pore-Scale Modeling (BG, BP, JOGMEC, Saudi Aramco, Shell, Statoil and Total) for their financial support.
Imperial College Consortium on Pore-Scale Modelling

References


Imperial College Consortium on Pore-Scale Modelling


Chapter 3

Modelling multiphase flow through micro-CT images of the pore space

Ali Qaseminejad Raeini

December 2010

Supervised by Dr. Branko Bijeljic and Prof. Martin Blunt
Abstract
Pore-scale modelling is a supplement to core analysis techniques to obtain a physical understanding of single and multiphase fluid flow through porous media, as well as to predict the macroscopic properties. The aim of this study is to develop a computationally efficient method to compute multiphase flow directly on micro-CT images of the pore space.

The incompressible Navier-Stokes equations are discretised by the finite volume method. The volume of fluid method is used to capture the fluid interfaces, and a Sharp Surface Force (SSF) model is devised based on the Continuous Surface Force (CSF) model for modelling surface tension forces. The SSF model is useful in eliminating the so called spurious currents present in the CSF formulation. Moreover, in order to develop a stable solution algorithm an additional equation is solved to calculate the capillary pressure. The calculated capillary pressure, after some corrections, is then used for consistent implementation of surface tension forces in the Navier-Stokes equations. These modifications make it possible to simulate multiphase flow in presence of large surface tension forces and complex solid boundaries in an Eulerian framework.

Several test cases are presented to show the accuracy of the numerical method. Some simulations for two-phase flow on micro-CT images of porous rocks are performed that will be used to compute the relative permeability and capillary pressure curves.
1 Introduction

1.1 Background

In many diverse geological and industrial areas, including petroleum reservoir engineering, environmental and chemical engineering, flows with two or even more phases through porous media are common practice. In petroleum reservoirs, multiphase flows are encountered in every stage of field development, including simultaneous flow of any pair of gas, oil and water, or sometimes solid particles towards production wells, as well as in many Enhanced Oil Recovery (EOR) processes. The analysis and understanding of multiphase flows is of paramount importance if processes involving multiphase flows are to be optimally and safely designed and controlled.

The principal objective of this study is to develop a robust methodology capable of direct numerical simulation of multiphase flow at the pore scale on micro-CT images of porous rocks, where the pore space contains millions of voxels, with sufficient accuracy and in a practical simulation time. This method will then be used to assess the variability in relative permeability and capillary pressure curves, which are commonly used in field-scale simulation of multiphase flow through porous media, and are the key indicators of the reservoir performance.

The remainder of this chapter is structured as follows: in the following section, the objectives of this work are stated. Afterwards, in section 1.3, the physical phenomena, controlling the behaviour of multiphase flow at the pore scale, are discussed. Next, section 1.4 reviews previous and related studies. In section 1.5, the literature review is followed by a list of the specific contributions made to the field. Finally, an outline of the contents of chapters 2 to 5 is given in section 1.6.

1.2 Objective of this research

The principal objective of this study is to develop a robust methodology capable of direct numerical simulation of multiphase flow at the pore scale on micro-CT images of porous rocks, where the pore space contains millions of voxels, with sufficient accuracy and in a practical simulation time. This method will then be used to assess the variability in relative permeability and capillary pressure curves, which are commonly used in field-scale simulation of multiphase flow through porous media, and are the key indicators of the reservoir performance.
1.3 Physical description of the problem

In this section an introduction of the physical phenomena controlling multiphase flow through porous media is presented. First, the physical laws describing the flow of two fluids at the pore scale are presented from a Computational Fluid Dynamics (CFD) point of view. Afterwards, the macroscopic laws used in field-scale simulation of multiphase flow are briefly discussed.

1.3.1 Microscopic description of pore space and the two-phase flow

Direct pore-scale simulation of multiphase flow requires a geometric representation of the pore space, on which the mathematical equations describing the state of the fluids are solved. In this section first an introduction of pore-space representation is given. Then we discuss the forces controlling the interaction of multiple fluids with each other and the solid rock.

1.3.1.1 Pore space representation

During the past decades, various methods have been developed to generate a three-dimensional (3D) representation of the pore space of a porous rock; see Dong and Blunt (2009) and references therein. They can be categorized into two groups. The first category includes various reconstruction methods to construct synthetic 3D rock images from high resolution 2D thin sections using statistical methods or geological process simulation. The second is direct imaging, which produces 3D images mapping the interior structure of the rock sample, such as the destructive approach of cutting and stacking serial 2D sections, confocal laser scanning microscopy and non-destructive X-ray micro-tomography (micro-CT). Among these methods, micro-CT technology provides a direct way to image the pore space into a high resolution volumetric (3D) representation of structures.

In this study we use the image files obtained from micro-CT imaging, which are in the form of “voxelized 3D binary images” formed from ‘0’s representing void spaces and ‘1’s representing the rock matrix. We convert the voxelized images directly to a mesh file of cubic grid blocks (see Figure 0.1), on which the Navier-Stokes equations are discretised and solved. This method has the advantage that the mesh structure is orthogonal and uniform. However, in this approach the pore walls will not be smooth. In addition, throats with small diameters, compared to image resolution, can not be accurately represented in a uniform Cartesian mesh (see Figure 0.2). An alternative approach is to convert the rock images using available mesh generation techniques to an unstructured mesh that has smooth solid walls. The disadvantage of this approach is that the mesh is non-orthogonal and non-uniform, and hence it will affect the accuracy of the numerical method.
Figure 0.2. A cross section of a coarse mesh, showing that small throats (in yellow) are not resolved well in a uniform Cartesian mesh.

In addition to more complex flow simulations on rock images, we also study simple idealized representations of pore-space, where theoretical solutions exist for the static or dynamics of the flow. This enables us to validate our method, and to analyse the accuracy of the numerical method that will then be applied for multiphase flow simulations on the rock images.

1.3.1.2 Surface tension forces and the force balance

At the nano-scale, surface tension forces are considered as a body force. A review of surface tension forces at these scales and their representation in terms of Lennard-Jones potential is given in Meakin and Tartakovsky (2009). However, at scales larger than nano-scales, they are usually considered as a surface force. A discussion of the surface tension forces as surface force is given next.

A volume element $V$ enclosing an interfacial surface $S$ is shown in Figure 0.3. As shown in this figure, surface tension forces ($\mathbf{t}_s$) are forces of magnitude $\sigma$ (the surface tension coefficient) per unit length. They act in the $s$ direction at every point along the closed contour $C$, defined as the cross-section of faces of volume element ($\mathbf{f}$) and the interface.

Performing a force balance on this volume element we get:

$$\int_V \rho \frac{D\mathbf{u}}{Dt} \, dV = \int_V \mathbf{f} \, dV + \int_{S_f} \mathbf{t}_f \, dS_f + \int_C \sigma \mathbf{s}_s \, dl$$  \hspace{1cm} (0.1)

where the term on the left hand side is the material derivative that accounts for the acceleration of the fluid within $V$ and $\mathbf{f}$ represents body forces. $\mathbf{t}_f = \mathbf{n}_f \cdot \mathbf{T}$ is the stress vector exerted on faces ($\mathbf{f}$) of the control volume, where $\mathbf{T} = -pI + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ is the stress tensor, $p$ is pressure, $\mathbf{u}$ is velocity vector and $I$ is the identity tensor. These terms are reformulated for Newtonian incompressible fluids in Chapter 2.
The term \( \int_C \sigma_s \, dl \) in Equation (0.1) accounts for the surface tension force exerted along the contour line \( C \) that has a length increment \( d\ell \). Brackbill et al. (1992) introduced the Continuous Surface Force (CSF) method, in which the capillary forces are approximated in terms of a body force \( \int_V f_{sv} \, dV \) acting on a narrow thickness of \([-H/2, H/2]\) around the interface:

\[
\int_V f_{sv} \, dV = \int_{s=H/2}^{H/2} \delta(n) \sigma (\nabla \cdot \mathbf{n}) \mathbf{n} \, dn \, dS,
\]

or:

\[
f_{sv} = \delta(n) \sigma (\nabla \cdot \mathbf{n}) \mathbf{n}, \tag{0.2}
\]

where \( \mathbf{n} \) is a parametrized coordinate normal to the interface \( S \), and \( \delta(n) \) should be chosen in a way that, in the limit of \( H \to 0 \), it takes the form of Dirac delta function.

Figure 0.4: Schematic representation of: (left) the transition region of the indicator function and (right) the indicator function versus the distance from the interface.

Brackbill et al. (1992) used \( \nabla \alpha \) to approximate \( \delta(n) \), where \( \alpha \) is a indicator function whose value changes smoothly from 1 in fluid 1 to 0 in fluid 2 over the transition region \([-H/2, H/2]\) (see Figure 3). This continuous formulation will be further discussed in chapter 3 for the discretisation of surface tension forces in the numerical method.

1.3.1.3 Solid wall adhesion effects (wettability)

Static contact angle:

At molecular scales, rock adhesion effects are produced as a result of uneven molecular forces in the contact line region. The behaviour of a fluid at contact line can be predicted by using molecular dynamics simulations or with other molecular-scale simulation methods such as phase-field model. In this study, however, we will consider only the macroscopic behaviour of the contact line.

Writing stress balance equations for a control volume of length scale \( \varepsilon^2 d\ell \) around a static contact line of length \( d\ell \) (see Figure 0.5), it can be observed that contact line and surface tension stresses scale as \( d\ell \), while the body (and acceleration) forces will scale as \( \varepsilon^2 d\ell \). Therefore, in the limit of \( \varepsilon \to 0 \), we have that the contact line stress must be balanced by the surface tension stress \( (\sigma_s d\ell) \). A direct consequence of this is that the interface normal in the vicinity of a smooth solid wall can be obtained from the static properties of the solid wall and the fluid system, i.e. using contact angle. Therefore the normal to the interface at the wall \( (\mathbf{n}_w) \) can be written as:

\[
\mathbf{n}_w \| = \cos \theta \mathbf{n}_w + \sin \theta \mathbf{s}_w \tag{0.3}
\]

where \( \theta \) is the static contact (wetting) angle, \( \mathbf{n}_w \) is the unit normal vector to the wall pointing towards the wall and \( \mathbf{s}_w \) is the unit vector tangential to the wall pointing towards the liquid and normal to the contact line. A value of \( \theta = 90^\circ \) means the interface is normal to the wall and \( \theta < 90^\circ \) indicates the fluid wets the wall.

January 2011 39
Contact angle hysteresis:
Due to heterogeneities, including structural (surface roughness) and chemical ones the contact angle can not be represented by a single static contact angle. It will depend on the history of the fluid distribution on solid walls, giving rise to contact angle hysteresis. If the scale of heterogeneities is larger than the size of grid-blocks used in the numerical scheme, the hysteresis effects can be predicted by the numerical method. In the case that the heterogeneities are smaller than the size of grid-blocks, the hysteresis effects will not be an outcome of the numerical method. In these cases, a dynamic contact angle should be used in the numerical method. In general the dynamic contact angle depends on the history of the fluid distributions over the solid surface. A recent literature review on the contact angle dynamics can be found in Meakin and Tartakovsky (2009, paragraphs 34-51).

Capillary pressure at the micro-scale
In static problems $f_c$ should have the form of a conservative force, hence it can be written in terms of the gradient of a scalar field $\nabla p_c$, namely microscopic capillary pressure, which can be calculated using the equation:

$$\nabla \cdot \nabla p_c = \nabla \cdot f_c, \quad \text{in } \Omega,$$

(0.4)

with the boundary condition:

$$\frac{\partial p_c}{\partial n} = f_c \cdot n \quad (= 0), \quad \text{on } \partial \Omega.$$

(0.5)

Together (0.4) and (0.5) define a Neumann problem for $p_c$ which has a unique solution up to an additive constant which can be obtained by fixing the value of $p_c$ at a reference point ($p_c(x_{ref}) = p_{c,ref}$).
This formulation for microscopic capillary pressure will later be used to filter the non-physical currents generated as a result of inaccurate representation of surface tension forces. In addition later in section 4.1, in order to validate the numerical scheme, the numerical solution of this equation will be compared to the theoretical capillary pressure for the case of modelling a static droplet as defined by Young-Laplace equation:

\[ \Delta p = \sigma \left( \frac{1}{R_1} + \frac{1}{R_2} \right) = \sigma K \]  \hspace{1cm} (0.6)

where \( K \) is the local mean curvature of the interface, and \( R_1 \) and \( R_2 \) are the principal radii of curvature.

### 1.3.2 Macroscopic description of porous media

At large scales, fluid flow simulations are usually based on the idea that the macroscopic properties of subsurface media, such as pressure and fluid velocities, can be described in terms of some average properties over a representative elementary volume (REV) scale sample, where REV-scale is the scale in which the fluctuations on smaller scales do not affect these macroscopic properties. For single-phase flow through porous media, fluid velocities then are related to microscopic pressure drop (\( \Delta \Phi / \Delta x \)) by the Darcy law (written in 1D for simplicity):

\[ u_d = \frac{q}{A} = \frac{k}{\mu} \frac{\Delta \Phi}{\Delta x}, \]  \hspace{1cm} (0.7)

where \( u_d \) is the Darcy velocity, \( q \) is the volumetric flux, \( A \) is the cross sectional area of porous material and \( \Phi \) is the flow potential. The REV concept is also applied broadly in the large-scale simulation of multiphase fluid flow, and the Darcy law is generalized to multiphase flow, relating flow rate of each phase (\( a \)) to the gradient of flow potential for that phase (\( \Delta \Phi_a / \Delta x \)):

\[ u_a = \frac{kk_{r,a}}{\mu_a} \frac{\Delta \Phi_a}{\Delta x}, \]  \hspace{1cm} (0.8)

where \( \Phi_a \) is defined as:

\[ \Phi_a = P_a - \rho_a g h - P_e, \]  \hspace{1cm} (0.9)

\( P_e \) is called macroscopic capillary pressure, which accounts for the surface tension effects, and \( k_{r,a} \) is the relative permeability. The macroscopic pressure for each phase (\( P_a \)) is usually defined as average of the microscopic pressures in that phase. Obtaining this average, however, poses a big challenge and it will be investigated in this research.

Equation (0.8) defines the three main macroscopic variables in multiphase flow: \( k_r \), \( P_e \) and \( \Phi \). It implies two trivial facts: (a) the flow rate of any phase at a specified saturation should depend linearly on the gradient of the potential \( \Phi \) (i.e. \( \Delta \Phi / \Delta x \)) and (b) the value of \( \Delta \Phi / \Delta x \) should be zero when the velocity field is zero. Knowing this: Capillary pressure difference is defined as the value of average pressure difference when the velocity goes to zero. Finally, the relative permeability of any phase, \( k_{r,a} \), is the ratio of flow rate of \( a \) in multiphase flow divided by the flow rate of that phase in the single-phase flow system (\( a = 1 \)) when the (dynamic) pressure drop, \( \Delta \Phi / \Delta x \), is the same for both systems:

\[ k_{r,a} = \frac{q_a \Delta \Phi_a}{q_{a=1} \Delta \Phi_{a=1}}, \]  \hspace{1cm} (0.10)
Relative permeability hysteresis: Porous media has heterogeneities of different types and at different scales; from chemical heterogeneities and roughness on solid walls of pores, to fluctuations in size and shape of pores and throats. This heterogeneous nature of porous media causes hysteresis in both $k_r$ and $P_c$. Figure 0.7 demonstrates a typical dependence of relative permeability and capillary pressure curves on the history of the saturations in porous media. Dependence of relative permeability hysteresis curves on various microscopic properties will be investigated in this study, using the results of multiphase flow simulations at the pore scale.

1.4 Previous related studies

During the past decades different methods have been developed that can be used for numerical simulation of multiphase flow problems. Each of these methods has their strengths and limitations. Among them, pore network models are now widely used for the simulation of flow in porous media. A brief discussion and literature review of pore network modelling is given in the next section. It is followed by a brief discussion of other methods, which are often referred to as direct methods, in section 1.4.2, and some of their advantages and limitations are addressed. In addition, we use finite volume method as a grid-based method which is widely used in modelling multiphase flow at industrial level as a CFD methodology. A more detailed review of grid-based methods is given in section 1.4.3.

1.4.1 Pore-network models

In pore network modelling, the void space of a porous rock is represented by a lattice of pores connected by throats. This lattice can be produced by various methods: In simple networks, regular networks may be used, with throats having circular cross section. More sophisticated networks can be produced by considering the spatial configuration and statistical data from real porous rocks or even by producing the network based on the real 3D images of porous rock. The cross-sectional shape of the throats can also be chosen so that it closely resembles the cross-sectional areas of throats in real rocks. The flow of different fluids is then modelled through the lattice by assuming certain assumptions on flow patterns, depending on the relative importance of capillary and viscous forces. In quasi-static models, it is assumed that the viscous forces are small compared to capillary forces. On the other hand, dynamic models consider both viscous and capillary force in the pore-network model.

Reviews of pore network models can be found in Blunt (2001); Blunt et al. (2002) for different types of pore network models applicable in modelling two-phase flow along with a review of applications of pore-network models, as well as in Piri and Blunt (2005) for three phase pore-network models, and also in Dong and Blunt (2009) where different methods for constructing a pore-network model are reviewed.

An important advantage of pore-network models over direct methods is their superior computational efficiency, which makes it possible to use them to model flow in large samples. Therefore, they can be used to analyse the effects of millimetre and centimetre heterogeneities in flow through porous media and serve as a bridge from the pore to the continuum scale. An example of such studies can be found in Balhoff et al. (2007); Idowu and Blunt (2010).

A good example of impact of pore-network modelling in our understanding of multiphase flow through porous media can be found in Nordbotten et al. (2008). They used the results of pore network models and
questioned the traditional macroscopic description of flow, and proposed a new formulation for macroscopic pressure to eliminate the limitations of traditional Darcy-type equations.

However, the superior efficiency of pore-network models comes with the expense of a significant loss in the physical detail that these networks can model. This is because of the fact that pore-network models are based on simplifications in geometry as well as physics of the problem, which adds to the uncertainty of the predicted results. In addition, in certain cases such as flow through fractures flow cannot be easily represented in terms of pore-network models (Ovaysi and Piri, 2010). These limitations of pore-network models bring us to the direct methods, which are reviewed in the following sections.

1.4.2 Direct methods
During past decades different methods have been developed for direct numerical simulation of multiphase flow problems. Some of these methods including (a) mesh free methods such as smoothed particle hydrodynamics (SPH) and moving particle semi-implicit (MPS) method, (b) lattice Boltzmann method (LBM) and (c) grid-based methods such as finite difference, finite element and finite volume methods can be used for the purpose of this study. Methods such as molecular dynamics, dissipative particle dynamics, lattice gas method and Monte Carlo methods are computationally expensive for solving flow in a core-size image, and they are not considered practical in this context. The applicability of some of these methods in modelling multiphase flow through porous media is reviewed in Meakin and Tartakovsky (2009). In the context of modelling multiphase flow through porous media, LBM (Ahrenholz et al., 2008; Hao and Cheng, 2010) and mesh free methods such as SPH (Tartakovsky et al., 2009) and MPS (Ovaysi and Piri, 2010), have been given special attention in recent years, which is mainly due to their flexibility in handling complex situations occurring in multiphase flow. However, there are some limitations attached to each of these methods.

The main limitation of the direct methods is their higher computational demand compared to pore-network models. The computational cost of the calculations is more pronounced in Lagrangian methods such as SPH and MPS; see for example Ovaysi and Piri (2010); Jiang et al. (2007). On the other hand, LBM has a superior numerical efficiency compared to Lagrangian mesh free methods. However, it has poor numerical stability and it is difficult to perform multiphase flow simulations under the wide range of density and viscosity ratios found in porous media (Meakin and Tartakovsky, 2009). In addition, when the capillary forces are treated as a continuous body force, it suffers from the presence of non-physical velocities near the interface (Pooley and Furtado, 2008).

1.4.3 Grid-based methods
Grid-based methods are widely used in industrial applications, which is mainly due to their ability to solve almost any partial differential equation accurately and in a practical simulation time. However, the main challenge in modelling multiphase flow using grid-based methods is the treatment of interface motion. Several methods have been developed during the past decades for modelling interface motion, applying the interface force and wettability effects in the Navier-Stokes equations. Extensive reviews of such methods for modelling interface motion can be found in Shyy (1996); Tryggvason et al. (2001); Scardovelli and Zaleski (1999); Hyman (1984). A brief review of some of these methods, including some of the recent advances, is given next.

1.4.3.1 Interface representation
There have been a variety of different methods developed to solve for the position of the interfaces. They can be classified into two main classes depending on the type of grids used: (a) moving mesh and (b) surface/volume tracking methods:

**Moving Mesh Methods:** Two phases are meshed separately, and the interface is represented as the boundary between two meshes. The mesh might be adjusted to fit the interface (see Figure 0.8a) or following the fluid (Lagrangian methods); see Rusche, (2002) and references therein.

**Surface/volume Tracking Methods:** Both fluids are meshed as they are a single fluid: The interfaces position is calculated by (a) marking and tracking the interface, either with a height function (Nichols and Hirt, 1973) or a set of marker particles/segments (fig. 6b) (Unverdi and Tryggvason, 1992; Popinet and Zaleski, 1999), or (b) the fluids in both sides of interface are marked either by massless particles (Harlow and Welch, 1965; Daly, 1967) or by an indicator function, which may be a volume fraction (fig. 6c) (Hirt and Nichols, 1981; Ubbink and Issa, 1999), a level set function (Sussman et al., 1994) or a phase-field variable (Nadiga and Zaleski, 1995; Jacqmin, 1999).
Moving mesh methods have the advantage that the interface is defined as a sharp boundary between two fluids. This approach allows a precise representation of the interfacial jumps conditions (Popinet and Zaleski, 1999). However, in the case of large interface motion, the mesh has to be relocated and eventually re-meshed, which adds to the complexity as well as the computational cost of the method. Surface tracking methods define a sharp interface, but the interface force is applied using the same ideas as volume tracking methods, as it will be discussed later in the next section. Another drawback of both surface tracking and moving mesh methods is that they require intervention to handle topological changes, e.g. the merging of two interfaces (Tryggvason et al., 2001). Moreover, surface tracking methods have the disadvantage that they do not strictly conserve the volume of each fluid (Tryggvason et al., 2001), whereas conservation can be enforced in some volume methods (when a volume fraction is used) and in some moving mesh methods.

In volume methods, an indicator function is used to represent the interface. There are three main approaches in choosing the indicator function: (a) The volume of fluid method uses a volume fraction of individual phases as the indicator function. (b) In the level-set method (LSM) the indicator function is the signed distance from the interface. (c) Finally, phase field model uses a phase field variable indicating the thermochemical state of different phases as the indicator function.

In LSM, since the indicator function used is a smooth function, calculation of curvature is more accurate. However, as advection deforms the signed distance function, it must be eventually reconstructed, which adds to the complexity of the solution procedure. In addition mass conservation is often significantly violated (Meakin and Tartakovsky, 2009). On the other hand, in the phase field model the nano-scale thickness of interface is considered in the mathematical model, hence it is more suitable for micro-scale simulations in predicting the behaviour of the interfaces and contact lines.

When the volume fraction is chosen as an indicator function, extra care should be taken to convect the interface without diffusing it. In addition, the convection scheme has to guarantee that the indicator function remains bounded between 0 and 1. In order to overcome these problems, two different methodologies to convect the volume fraction have been developed: in volume-of-fluid (VOF) methods the interface is reconstructed from the volume fraction distribution before advecting it (Hirt and Nichols, 1981; Ashgriz and Poo, 1991; Rudman, 1997), whereas in interface-capturing techniques high-order convection schemes are used to guarantee the boundedness of the indicator function and at the same time maintain a sharp interface (Davis, 1994; Ubbink, 1997; Rusche, 2002).

1.4.3.2 Representation of surface tension forces

The surface tension term in the Navier-Stokes equation creates additional difficulties because it is a singular term. These difficulties manifest themselves in terms of numerical instabilities and/or numerical noise as well as poor representation of capillary effects. In moving mesh methods, the surface tension force can be applied as a pressure jump across the interface, as the interface is aligned with the mesh surfaces and serves as a boundary between two fluids (Scardovelli and Zaleski, 1999). On the other hand, in surface tracking and volume methods, the interface force should be applied in an Eulerian mesh which is independent from the interface and therefore it needs special treatments.

Recently, different methods are developed to apply the surface tension force in the Eulerian grid; such as: (a) Continuous Surface Stress (CSS) which is mathematically equivalent to (b) Continuous Surface Force (CSF) method (Brackbill et al., 1992; Lafaurie et al., 1994), and finally (c) Sharp Surface Force (SSF) (Renardy and Renardy, 2002) also named Ghost Fluid Model (GFM) (Francois et al., 2006).
An often reported problem is the existence of the so-called spurious currents in the flow field of the numerical simulations (Brackbill et al., 1992; Lafaurie et al., 1994; Brackbill and Kothe, 1996; Popinet and Zaleski, 1999; Williams et al., 1998; Ubbink, 1997; Renardy and Renardy, 2002; Francois et al., 2006; Dupont and Legendre, 2010). These currents are vortices which appear in the neighbourhood of interfaces despite the absence of any external forces. Lafaurie et al. (1994) implemented a conserving form of the CSF. He showed that the magnitude of the largest spurious current \( U_s \) around a bubble/droplet can be given as:

\[
U_s = K \frac{\sigma}{\nu}
\]  

(0.11)

Here, \( \nu \) and \( \sigma \) are the dynamic viscosity and surface tension, respectively. They report that \( K \) is of the order of \( 10^{-2} \) in their method.

Brackbill and Kothe (1996) derived a CSF-like formulation from an energy functional. They suggests that spurious currents are best eliminated by controlling the interface thickness. Renardy and Renardy (2002) implemented a Parabolic Reconstruction Of Surface Tension (PROST) method that reduced the magnitude of the spurious currents. More recently, Francois et al. (2006) have developed the Sharp Surface Force (SSF) method that reduced the magnitude of the spurious currents. In these studies, Renardy and Renardy (2002) and Francois et al. (2006) suggested that the spurious currents can be reduced by implementing a more accurate method for calculation of interface curvature.

Equation (0.11) implies that the problem of spurious currents is more pronounced at low capillary numbers. Therefore, it can be considered as a main limitation when modelling multiphase flow in porous media, where capillary number is low (roughly in the range \( 10^{-10} \) to \( 10^{-4} \)).

1.4.3.3 Contact line and contact angle dynamics

The wetting of a solid by a liquid remains only partially understood (Dussan, 1979; de Gennes, 1985; Blake, 2006; Dupont and Legendre, 2010). The description of the contact line is complicated due to two main reasons: (a) The Navier-Stokes equations with standard no-slip boundary conditions produce an infinite viscous dissipation (Huh and Scriven, 1971). This particular complexity of the physics of the contact line makes the numerical simulations very delicate (Dupont and Legendre, 2010). (b) The solid walls have heterogeneities at different scales, which affect the dynamics of contact line. Therefore, the physics of contact line can be modelled only in certain molecular scale simulation methods (Blake, 2006).

However, in CFD applications the numerical method solves the flow to describe the large scale behaviour of the interface. Therefore, proper boundary conditions should be developed so that the numerical method can predict the large-scale behaviour of the contact-line, without directly incorporating the nano-scale (Van der Waals) interactions between the fluids and the solid wall. In summary, the numerical model should be able to predict the macroscopic behaviour of (a) contact line statics, (b) the stick-slip behaviour of contact line observed in transition between static and motion, as well as (c) the motion of the receding and advancing contact lines.

Recently different methods have been used for describing the dynamics of contact lines in numerical models; see (Spelt, 2005) and (Dupont and Legendre, 2010) for recent reviews on the subject.

In the VOF method, Renardy et al. (2001) investigated two different methods to implement the macroscopic effects of contact angle for smooth solid boundaries; the first method treats the problem as a three-phase situation and mimics the classical argument of Young. The second method modifies the interface normal, which are used in calculation of curvature, at solid boundaries so that they represent the contact angle. They concluded that the second approach gives better results. Huang et al. (2005) used the VOF method to simulate two-phase flow in a 2-D rectilinear finite difference grid. They reported that the common methods that are used for implementing contact angle on smooth surfaces can lead to large spurious current. They used a stair-case method for representing the complex solid boundaries that, as they report, alleviate the problem of these non-physical currents.

1.5 Novel Contributions in this study

1.5.1 CFD methodology

We use OpenFOAM (2009a) as an open-source package for solving the governing differential equations, which provides us with different utilities for pre/post-processing, different numerical schemes for discretisation of differential equations and a number of high performance sparse linear equation solvers. Moreover, OpenFOAM has a two-phase flow solver, called interFoam (OpenFOAM, 2009a; Ubbink, 1997; Rusche, 2002), to which we made several modifications to make it suitable for simulation of two-phase flows at the pore scale with large
surface tension forces and irregular solid boundaries; we call this code iInterFoam solver in this study. The following new methods and modifications are implemented in the iInterFoam solver, during this study:

1. A new sharp surface force (SSF) method is used instead of CSF, in order to eliminate the spurious currents in modelling free surfaces, as discussed in section 3.3.

2. The projection-correction algorithm is modified by defining a new dynamic pressure and using it in correction step, rather than using total pressure (static + dynamic) in the correction step (see section 2.2). This makes it possible to filter non-physical currents generated close to curved/angular solid boundaries, which are produced as a result of inaccurate calculation of capillary forces. In addition it helps us to filter the non-physical net-force applied on a moving droplet due to numerical errors in calculation of capillary forces.

3. The calculation of interface curvature is modified to produce a more accurate calculation of capillary pressure.

4. A new contact angle hysteresis model will be developed to add support for the physical phenomena such as stick-slip behaviour of the contact line, in modelling contact line movement at micro-scales.

These modifications in the numerical formulation make the numerical method more accurate and stable for solving multiphase flow problems for low capillary numbers, and eliminate the spurious current problem in the original CSF formulation.

1.5.2 Pore-scale methodology

We will use the iInterFoam code as a grid-based multiphase flow solver using VOF method as an efficient and stable solver for direct numerical simulation of multiphase flow at the pore scale. It will be used to study different processes occurring in porous media, such as drainage and imbibition, as well as to obtain the macroscopic properties defined for each of these processes, e.g. relative permeability and capillary pressure curves.

The improved stability of this code makes it suitable for extending it for modelling more complicated phenomena, such as phase change and chemical reaction, in the presence of surface tension forces.

Direct Numerical Simulation of Multiphase Flow at the Pore Scale (DNSMFPS) incorporates more fundamental properties, such as contact angle and pore geometry. Consequently they offer advantages in analysing laboratory experiments, compared to macroscopic Darcy-type laws which use average parameters (capillary pressure and relative permeability), for describing multiphase flow in porous media. Therefore DNSMFPS can help in understanding the physical phenomena occurring at porous media more thoroughly, which helps in choosing better strategies in the process designs in environmental and petroleum engineering applications.

1.6 Outline of the report

Chapter 2 discusses the mathematical model and the governing equations describing multiphase flow at pore scale. The numerical solution will be presented in chapter 3. In chapter 4 several test cases are presented and the results are compared against the theoretical solutions and experimental measurements. The goal of this chapter is to validate the accuracy of the proposed numerical method. Finally, chapter 5 discusses the future work for the rest of this Ph.D. research.
2 Mathematical description

In this section, the mathematical equations describing the flow of multiple fluids are summarized. The mathematical formulations are based on a single-fluid methodology, i.e. one single momentum conservation equation is solved for both fluids. More discussion about the equations presented here can be found in Ubbink (1997) and Rusche (2002).

2.1 Governing equations

2.1.1 Mass and momentum conservation

The mass and momentum conservation equations for the incompressible fluids are as follows:

\[ \nabla \cdot \mathbf{u} = 0 \]  
\[ \frac{D}{Dt}(\rho \mathbf{u}) = \nabla \cdot \mathbf{T} - \nabla p + \mathbf{f}, \]  

where \( \mathbf{u} \) is the velocity vector, \( p \) is the pressure, \( \mathbf{T} \) is the stress tensor. For the case of incompressible flow \( \nabla \cdot \mathbf{T} \) can be written as:

\[ \nabla \cdot \mathbf{T} = \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \mathbf{u} \cdot \nabla \mu, \]  

\( f = f_g + f_c \) represents all external forces; \( f_g = \rho g \) is the gravity force and \( f_c \) is the surface tension force, as defined in equation (2.6). Finally, \( \rho \) and \( \mu \) are average fluid density and viscosity, which are calculated using the indicator function:

\[ \rho = \alpha \rho_1 + (1 - \alpha) \rho_2 \]  
\[ \mu = \alpha \mu_1 + (1 - \alpha) \mu_2, \]  

where \( \alpha \) is the indicator function that is discussed in more detail in the next section.

2.1.2 Advection of the indicator function

In the volume-of-fluid method, the interface is represented by an indicator function (\( \alpha \)) which represents the volume fraction of one of the fluids in each grid cell. If the cell is completely filled with the first fluid then \( \alpha = 1 \) and if it is filled with the second fluid \( \alpha \) should be 0. At the interface the value of \( \alpha \) is between 0 and 1. The volume fraction \( \alpha \) of the fluids is evolved with an advection equation of the form:

\[ \frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \mathbf{u}) = 0 \]  

As it was discussed in Section 1.3.1.2, surface tension force (\( f_c \)) is computed as a body force,

\[ f_c = \sigma k \mathbf{n}_i \delta_j \]  

where \( k = \nabla \cdot (\mathbf{n}_i) \) is the interface curvature and \( \mathbf{n}_i \) is the normal to the interface:

\[ \mathbf{n}_i = \frac{\nabla \alpha}{|\nabla \alpha|} \]  

Finally, \( \delta_j \) is a delta function concentrated on the interface. In CSF formulation (Brackbill et al., 1992) the indicator function, \( \alpha \), is considered as a smooth continuous function. The term \( \mathbf{n}_i \delta_j \) is then approximated as:

\[ \mathbf{n}_i \delta_j = \nabla \alpha. \]  

2.2 Treatment of singular capillary pressure jumps

The CSF formulation for surface tension forces, eq. (2.8) is not suitable for numerical modelling of surface tension forces in the case of capillary dominated flow regimes. In these cases the capillary forces are significantly higher than viscous forces and any numerical error in calculation of \( \nabla \alpha \) will cause high spurious velocities. Therefore eq. (2.8) is not used in the numerical method. Instead, we use the modified form of this equation as discussed in section 3.3.1.
In addition, in order to avoid the difficulties in discretisation of the singular term \( f_c \), the momentum equation (eq. (2.2)) is rewritten in terms of the microscopic capillary pressure \( p_c \) and a dynamic pressure \( p_d = p - p_c \) as follows:

\[
\frac{D}{Dt}(\rho u) = \nabla \cdot \mathbf{T} - \nabla p_d + f', 
\]

where

\[
f' = \rho g + f_c - \nabla p_c
\]

This approach for including the effect of capillary forces in the Navier-Stokes equations enables us to filter the numerical errors related to inaccurate implementation of \( f_c \) as it will be discussed later in chapter 4. In addition, it provides a consistent way of calculating Darcy-scale dynamic and capillary pressure, in order to analyse the macroscopic behaviour of flow through porous media.
3 Numerical method

In this chapter a brief discussion of the application of the finite volume method in discretisation of the partial differential equations (PDEs), presented in previous chapter, is given. More detailed discussions about the finite volume discretisation procedure, which is used in this study, can be found in Jasak (1996); Ubbink (1997); Rusche (2002); OpenFOAM (2009b). A discussion of the finite volume discretisation practice is given in Appendix 0, for the sake of completeness.

In finite volume discretisation the solution domain is subdivided into small intervals, forming volume elements for space domain and time intervals for the time domain. The continuum variables are then replaced by collocated variables at specified locations in the space and time domain. By convention, in this study, all primary variables (specifically velocities, pressure and indicator function) are defined at the centre of volume elements, which are referred to as volume fields in this report. If the values of such variables are needed at face centres, they will be calculated using interpolation of cell-centred field using appropriate schemes as discussed in Appendix 0.

Time intervals are handled differently. By convention, we define the velocity field to be defined at the edges of time intervals while pressures and indicator function are assumed to be collocated to the centres of the time intervals. This choice of collocation is used to make the numerical method efficient while maintaining the accuracy, as it will be pointed out later in this chapter.

In general, equation discretisation converts the PDEs into a set of algebraic equations (SAEs). These SAEs will be solved using appropriate numerical methods to obtain a solution for the set of unknown collocated variables $\phi$. SAEs are commonly linearised and expressed in the matrix form as:

$$ \sum_i A_i(\phi) = 0, \quad (3.1) $$

where $A_i$ can be any integro-differential operator. The discretised form of this PDE can be linearised and expressed in the matrix form as:

$$ \sum_i ([A_i] \phi + [b_i]) = [0] \quad (3.2) $$

where $[A_i]$ and $[b_i]$ are the matrices of coefficients and source vectors corresponding to the discretisation of $A_i$ s, respectively. In this report we use the same notation for the discretised operators as those of the continuum differential operators, $A_i$ s. Whether the discretised operator has contributions in $[A_i]$, $[b_i]$ or both will hopefully be clear from the context. Some of the discretised operators used in this study are briefly discussed in Appendix 0. A volume element is also shown in Figure 3.1, in which the discretisation parameters used in this study are shown graphically.

![Figure 3.1: A volume element, and the parameters used in finite volume discretisation (from Rusche (2002)).](image)

3.1 Discretisation of the equation for advection of indicator function

The indicator function has the form of a step function at continuum limit, while the numerical algorithm tends to smear the interface sharpness. Therefore, special attention is needed in discretisation of this advection equation. In the numerical method used in this study an extra artificial compression term is introduced into the VOF equation (eq. (2.5)), in order to control the thickness of the interface (see Rusche, 2002) for a discussion on the subject). The discretised form of advection of $\alpha$, after implementing the artificial compression term, reads:
\[
\frac{\partial}{\partial t} \alpha + \nabla \cdot (\alpha \mathbf{u}) + C_\alpha \nabla \cdot [\alpha (1 - \alpha) \mathbf{u}_f] = 0,
\]
(3.3)

or

\[
\frac{\partial}{\partial t} \alpha + \sum_{j = S_f} \left( \langle \alpha \rangle_f \phi + C_\alpha \langle \alpha \rangle_f (1 - \alpha) \phi_i \right) = 0,
\]
(3.4)

where \( \phi = \mathbf{u}_f \cdot S_f \) is the volumetric flux as it will be discussed later in this chapter. \( \phi_i = (\mathbf{u}_i \cdot S_f) \) is defined as:

\[
\phi_i = |\phi| \langle \mathbf{n}_i \rangle_f \cdot \mathbf{n}_f.
\]
(3.5)

This artificial term is active only in the interface region due to the term \( \alpha (1 - \alpha) \). \( \langle \cdot \rangle_f \) is used to denote the face-centred fields that are calculated by interpolating the corresponding cell-centred field as discussed in Appendix 0. Linear interpolation scheme (i.e. using equation (0.14)) is used for interpolation of \( \langle \mathbf{n}_i \rangle_f \) in this equation. Our simulations show that the scheme used for interpolation of \( \langle 1 - \alpha \rangle_f \) and \( \langle \alpha \rangle_f \) affects the results significantly. In most of our simulations the SFCD scheme (Ziman, 1990) as adopted for finite volume calculations by Jasak (1996) is used for this purpose.

The time derivative can be discretised using a variety of schemes. The Euler scheme (eq. (0.4)) is used for discretisation of the temporal derivative and explicit scheme (using \( \phi \) from the previous time step) is used for discretisation of the advection terms.

### 3.2 Updating interface curvature

Once the indicator function is advected, interface normals and the interface curvature should be updated to calculate the surface tension forces. Interface normals will be calculated using the discretised form of equation (2.7):

\[
\mathbf{n}_i = \frac{\nabla \alpha}{(|\nabla \alpha| + \varepsilon)},
\]
(3.6)

where \( \varepsilon \) is a small number compared to the values of \( |\nabla \alpha| \) at the interface region. The Gauss-linear scheme (eq. (0.22) and a linear interpolation to obtain the face-centred field \( \langle \alpha \rangle_f \) from the cell-centred field \( \alpha \) ) is used for calculation of the \( \nabla \alpha \):

\[
\nabla \alpha = \sum_{j \in S_f} \frac{1}{V_j} \langle \alpha \rangle_f S_j
\]
(3.7)

To incorporate the effect of wall adhesion into this equation, the interface normals at solid boundaries are modified so that they represent the contact angle. Therefore, the face-centred vector field \( \mathbf{n}_{sf} \) is calculated as follows:

\[
\mathbf{n}_{sf} = \begin{cases} 
\langle \mathbf{n}_i \rangle_f & \text{if } f \in \text{internal faces} \\
\mathbf{n}_i |_{w} & \text{if } f \in \text{solid walls}
\end{cases}
\]
(3.8)

where \( \langle \mathbf{n}_i \rangle_f \) is calculated using a linear interpolation scheme from the cell-centred value defined by equation (3.6) and \( \mathbf{n}_i |_{w} \) is obtained using equation (0.3). The face-centred values of interface normals, \( \mathbf{n}_{sf} \), are then used to calculate the interface curvature at cell centres, using the Gauss scheme:

\[
k = \nabla \cdot (\mathbf{n}_{sf}) = \sum_{j \in S_f} \frac{1}{V_j} S_f \cdot \mathbf{n}_{sf}
\]
(3.9)

The curvature obtained by this method is valid for those cells where the gradient of the indicator function is not zero, which means for those cells at the interface region and their nearest neighbours.

### 3.3 Surface tension force

Capillary pressure \( p_c \) is a discontinuous function and the terms \( \nabla p_c \) and \( f_c \) are singular. The original VOF method as presented by Brackbill et al. (1992) considers these terms as a smooth, continuous and differentiable variables. However there are some major deficiencies in the original VOF method, which may result in unacceptable errors especially when capillary forces are high compared to viscous forces.
In the rest of this section we present a stable method for calculation of surface tension forces that are used in the capillary pressure equation (eq. (0.4)), which in turn, is used for proper implementation of projection method in solving the Navier-Stokes equations.

### 3.3.1 Sharp surface force model

In order to make it possible that surface tension forces be exactly balanced by pressure gradient (see Francois et al. (2006)), the surface tension force is directly calculated at face centres:

\[
\mathbf{f}_{s_f} = \mathbf{f}_{c,f} \cdot \mathbf{n}_f = \langle \sigma_k \mathbf{n} \delta \rangle_f \cdot \mathbf{n}_f = \sigma(k) \delta_f
\]  

(3.10)

In order to control the sharpness of capillary pressure, the interface delta function \( \delta_f \) is calculated from a sharpened indicator function \( \alpha_{pc} \) as follows:

\[
\delta_f = \nabla_j^+(\alpha_{pc})
\]  

(3.11)

where \( \nabla_j^+(\alpha_{pc}) \) is calculated using equation (0.18). \( \alpha_{pc} \) is the modified indicator function, which is obtained by curtailing and then normalizing \( \alpha \) as follows:

\[
\alpha_{pc} = -\frac{1}{2 C_{pc}} \min\{ \max(\alpha, C_{pc}), 1 - C_{pc} \}
\]  

(3.12)

A value of \( C_{pc} = 0 \) leads to the original CSF formulation, while as \( C_{pc} \) approaches 0.5, \( \alpha_{pc} \) becomes sharper, and hence a sharper implementation of the surface tension forces will be achieved. The appropriate choice for \( C_{pc} \) will be discussed in Chapter 4. A value of 0.49 was observed to give the best results for static problems, in terms of elimination of spurious currents. For this choice of \( C_{pc} \) the capillary pressure transition area will effectively be limited to the distance between two grid-blocks, and hence the term Sharp Surface Force (SSF) model is used for this formulation. A graphical representation of the effect of this technique for sharpening the capillary force is given in Figure 3.2.

![Figure 3.2: Graphical representation of the coefficient \( C_{pc} \) used for controlling the sharpness of capillary pressure:](image)

(a) The sharpened indicator function \( \alpha_{pc} \) as a function of \( \alpha \) and (b) An illustration of the shape of \( \alpha_{pc} \) and \( \alpha \) as a function of distance \( n \) from the interface.

### 3.3.2 Discretisation of the static capillary pressure equation

The discretised form of the capillary pressure equation (eq. (0.4)) can be written as follows:

\[
\nabla \cdot \nabla p_c = \nabla \cdot \mathbf{f}_c , \quad \text{in } \Omega,
\]  

(3.13)

where \( \nabla_j^+ p_c \) is calculated using equation (0.18). In this formulation both capillary pressure gradient and capillary forces are calculated directly at the required (face-centred) location, not by interpolating the forces from another location (cell centres) to the location of interest (face centres). This algorithm ensures that capillary...
pressure gradient is balanced exactly by the capillary force, and hence it is referred to as “balanced-force algorithm” in Francois et al. (2006).

The SSF formulation in conjunction with the balanced-force algorithm is helpful in eliminating the non-physical behaviour in absence of complex solid boundaries. However, the preliminary results show that the formulation given above results in non-physical currents close to complex (curved or edged) solid boundaries that are parallel to fluid interfaces. To eliminate this problem, the components of surface tension forces that result in currents parallel to fluid interfaces are filtered. This filtering technique and its effects on numerical results are presented in section 4.3.

3.4 Discretisation of the momentum equation

In this section the discretisation of momentum and continuity equations are presented. More details about the discretisation practice which are not covered here can be found in Ubbink (1997, pages 23-26) and Rusche (2002, chapters 2 and 4). The focus in this section will be to present the modifications made in this study to resolve the problems associated with high capillary pressures.

The discretisation of momentum equation (eq. (2.9)) is obtained in a similar method as advection of indicator function, i.e. by taking the integral of the momentum equation over the time interval and cell volume, as discussed in Appendix 0, resulting in:

\[
\frac{d}{dt}(\rho u) + \nabla \cdot (\rho uu) - \nabla \cdot (\mu \nabla u) = \nabla u \cdot \nabla \mu - \nabla p_d + f',
\]

This equation represents the conservation of a vector quantity, which can be broken into the conservation of each of its components, thus in 3D space three separate systems of linear equations are obtained. In this study, the discretised time derivative \(\frac{d}{dt}\) is obtained by the Euler method. The convection term \(\nabla \cdot (\rho uu)\) is discretised using equation (0.20), by substituting \(\phi\) with \(u\). The Laplacian term \(\nabla \cdot (\mu \nabla u)\) is discretised using Gauss scheme (eq. (0.17)). The terms \(\nabla u\) and \(\nabla \mu\) are discretised using Gauss scheme in an explicit manner using equation (0.22). With these choice of discretisation schemes, the final set of discretised equations can be written as:

\[
\frac{\rho u^n}{\Delta t} + \sum_{j \in S_i} \frac{1}{v_i} \rho \phi^{j^n} + \sum_{j \in S_i} \frac{1}{v_i} \mu_j \left| \frac{u^n - u_n^b}{d} \right| = S_u - \nabla p_d + f',
\]

where \(\phi = S_j \cdot (u)_j\) is the volumetric flux and:

\[
S_u = \frac{\rho u}{\Delta t} + \sum_{j \in S_i} \frac{1}{v_i} S_j (u)_j \cdot \sum_{j \in S_i} \frac{1}{v_i} \mu_j
\]

For simplicity of the notation the superscripts ‘\(n-1\)’ and ‘\(i\)’ (counter for grid-cells) are dropped from all of the time-dependent variables. All the face velocities on the left hand side of eq. (3.15), for each cell, can be written in terms of the velocities at the centres of the cell and its nearest neighbours. It is possible to reformulate this equation in terms of these values:

\[
A u^n = \sum_{nb} A_{nb} u_{nb}^n + S_u - \nabla p_d + f'
\]

where the subscript \(nb\) counts for all of the nearest neighbours of each cell. It is useful to present the discretised momentum equations (eq. (3.17)) as:

\[
u^n = \nu^* - \frac{1}{A} (\nabla p_d)
\]

where:

\[
u^* = \frac{H(u) + f'}{A},
\]

and

\[
H(u) = \sum_{nb} A_{nb} u_{nb}^n + S_u.
\]

This formulation of the discretised momentum equations will be used for the derivation of the dynamic pressure equation, which follows in the next section.
3.5 Dynamic pressure equation

The pressure equation is derived from the discretisation of the incompressibility condition, eq. (2.1), and the discretised momentum equation (3.18). The finite volume discretisation of the incompressibility condition can be written as:

\[ \nabla \cdot \mathbf{u} = \frac{1}{v_i} \sum_{j \in S_i} \mathbf{S}_j \cdot \mathbf{u}_f = \frac{1}{v_i} \sum_{j \in S_i} \phi = 0. \quad (3.21) \]

In this equation, the discretised momentum equation (3.18) is used to predict the face values of the velocity, \( \mathbf{u}_f \). However, special care needs to be taken in obtaining the face-centred velocity from the interpolation of the cell values, in order to avoid decoupling of the velocity and pressure fields. The face interpolation proposed by Rhie and Chow (1983) is adopted in the present study. This is done by isolating the contribution of the pressure gradient and body forces from eq. (3.18) when interpolating them to the face. The contribution of the pressure gradients and body forces at the face is then added explicitly to \( \mathbf{u}_f \) by calculating them directly at the faces (Rusche, 2002). Consequently the face-centred flux field reads:

\[ \phi = \mathbf{u}_f^* \cdot \mathbf{S}_f + \frac{1}{\langle A \rangle_f} (\nabla^T \mathbf{p}_d) \mathbf{S}_f, \quad (3.22) \]

where \( \mathbf{u}_f^* \) is predicted as follows:

\[ \mathbf{u}_f^* = \frac{\langle \mathbf{H(u)} \rangle_f + \mathbf{f}_f}{\langle A \rangle_f}, \quad (3.23) \]

using the last known values of \( \mathbf{u} \). Replacing \( \phi \) from equation (3.22) into equation (3.21), the pressure equation will be:

\[ \nabla \cdot \left( \frac{1}{\langle A \rangle_f} \nabla^T \mathbf{p}_d \right) = \nabla \cdot \left( \mathbf{u}_f^* \right) \quad (3.24) \]

3.6 Solution algorithms

3.6.1 Pressure-velocity coupling

The velocity and pressure have a strongly linear coupling which has been a research topic for several years (Patankar and Spalding, 1972; Issa, 1986), and several algorithms have been developed for iterative solution of pressure and velocity equation. The pressure-velocity coupling algorithm used in this study is based on the Pressure Implicit with Splitting of Operators (PISO) algorithm of Issa (1986). The algorithm can be summarised as follows:

1. **Estimate the non-solenoidal velocity field**, \( \mathbf{u}_f^* \): use equation (3.23) to obtain a velocity field, which do not have the effect of pressure gradient.

2. **Pressure equation**: solve equation (3.24) for dynamic pressure using the estimated non-solenoidal velocity field from previous step.

3. **Momentum corrector**: use equations (3.18) and (3.22) to correct the velocity and face flux (\( \phi \)) fields, respectively, using the calculated pressure field from step 2. Repeat the whole PISO loop if necessary.

3.6.2 Solution procedure for the multi-fluid system

In order to set up the equation for prediction of the momentum, (3.19), the new densities, viscosities and the interface curvature (\( k \)) are needed. These are obtained using equation (2.4) to update \( \mu \) and \( \rho \) and using equation (3.9) to calculate \( k \). In all these equations the values of the indicator function at the centre of time interval are needed. Therefore the equation for advection of indicator function, eq. (3.3), should be solved before solving the momentum and pressure equation. For this purpose, the volumetric fluxes (\( \phi \)) from the end of last time interval are used in the equation (3.3). The final solution procedure for the multi-fluid system is shown in Figure 3.3.
Figure 3.3: Solution algorithm.

Start

Update fluid interfaces (solve eq. (3.4) for $\alpha$)

Calculate: viscosity, $\mu$, and density, $\rho$, (eq. (2.5)), interface curvature, $k$, (eq. (3.8)) and capillary force $f_{c,j}$ (eq. (3.10))

Solve eq. (3.13) for capillary pressure, $p_c$

Set up momentum equation (eq. (3.17))

Calculate intermediate velocity, $\mathbf{u}_i^j$ (eq. (3.23))

Solve dynamic pressure, $p_d$, equation (eq. (3.24))

Calculate $\mathbf{u}$ and $\phi$ from $\mathbf{u}_i^j$ and new dynamic pressure field (eqs. (3.18) and (3.23) respectively)

End
4 Test cases

In this chapter several test cases are presented to validate the accuracy of the numerical method presented in the previous chapter. As a general statement, our simulations show that the following two criteria are important in numerical modelling of multiphase flow at pore-scale:

(a) controlling the sharpness of capillary pressure, in order to eliminate spurious currents.

(b) maintaining the constraint of a “zero net capillary force” on a closed interface: \( \oint \mathbf{f}_c \cdot \mathbf{S}_i = 0 \) This can be achieved by using a more accurate and stable method for the calculation of surface tension forces.

In the numerical method presented in this study, the sharpness of capillary pressure is controlled by the term \( C_{pc} \), used in discretisation of the capillary pressure equation (eq. (3.12)).

However, it is difficult to maintain the zero net capillary force constraint \( \oint \mathbf{f}_c \cdot \mathbf{S}_i = 0 \). This behaviour was observed in modelling the steady movement of a micro-scale droplet in a capillary tube, both in CSF and SSF formulations. In CSF formulation the results are affected both by spurious currents as well as well as violation of this constraint. In SSF formulation, however, the interface tends to tend to attach itself to the mesh and stays stationary, when the capillary number is low. This problem can be alleviated by using a more accurate evaluation of surface tension forces.

The errors in surface tension forces are directly related to the calculation of the interface curvature as well as to the method used for approximation of interface delta function, \( \mathbf{n} \delta_\gamma \). Interface curvature itself is calculated from the gradient of indicator function, hence its accuracy is affected by the smoothness of indicator function. Therefore one approach for improving the accuracy of calculated curvature is to smooth the indicator function before calculating the curvature.

Since it is practically impossible to decrease the errors in calculation of capillary forces to zero, consequently it is difficult to model the movement of fluid interfaces accurately at low capillary numbers, where the capillary forces are significantly larger than the viscous forces and therefore the violation of zero-net capillary force plays an important role. Therefore, filtering the non-physical fluxes generated due to the inconsistent calculation of surface tension forces is necessary in maintaining the zero-net capillary force.

In the following sections we investigate the effect of some of the simulation parameters which are important for the accuracy of the simulation results, specifically in satisfying the two criteria discussed above. For this purpose, four test cases are selected: (a) a static 3D droplet, where we study the effects of mesh resolution, time-step size, sharpness of capillary pressure, sharpness of indicator function and smoothing the indicator function for calculation of the interface curvature, (b) a droplet moving with constant velocity, where we study the filtering of capillary forces in order to enforce the zero net capillary force constraint, (c) resting of a droplet on a flat surface with different contact angles to study the the accuracy of the numerical method in modelling the statics of contact angle, and (d) capillary rise in a circular tube, where we study the dynamics of contact angle.

4.1 Static droplet

In this section, the numerical results for modelling the behaviour of a 3D droplet of a liquid suspended in another liquid are presented. The flow domain is a cube with a size of \( 100^3 \mu m \). A cubic block of water (fluid 1) of size \( 20^3 \mu m \) is located in the centre of this domain, at room temperature and pressure. The rest is filled with another fluid with the same properties as those of the water, having a surface tension of 0.07 \( N/m \) with fluid 1. This means there is no buoyancy. The solution domain is shown in Figure 4.1. Boundary conditions are zero-gradient (Neumann) for both \( p \) and \( \alpha \), and no-slip for velocity.
Figure 4.1: Flow domain for modelling a static droplet: (a) initial condition for $\alpha = 0$, (b) final $\alpha = 0.5$ contour-surface for a mesh of resolution $20^3$.

The initial cubic block of water will be deformed as a result of surface tension forces, producing some capillary waves which will be gradually dampened due to the viscous forces. The final solution for the problem is that the velocity should be zero in the entire flow domain and the pressure is given by the Young-Laplace equation:

$$\Delta p_c = 2\sigma \frac{R}{R} = \frac{2 \times 0.07 \text{ N/m}}{0.000025 \text{ m}} = 5642 \text{ Pa}, \quad (4.1)$$

where $R$ is the droplet radius. This case is used to investigate the effect of sharpness of the capillary pressure, mesh resolution, time step size and the smoothing indicator function for calculation of the interface curvature.

4.1.1 Sharpening capillary force: CSF versus SSF formulation

The original CSF formulation proposed by Brackbill et al. (1992) produce spurious velocities near the interface, evidence of which can be found even in recent studies such as (Renardy and Renardy, 2002; Francois et al., 2006; Dupont and Legendre, 2010).

As suggested by Brackbill and Kothe (1996) the sharpness of interface forces is the key in eliminating spurious currents. In the numerical method used in this study the sharpness of indicator function is controlled by the $C_{pc}$ coefficient, used in equation (3.3). A sharper representation of capillary pressure can be achieved by choosing the value of $C_{pc}$ close to (but less than) 0.5 (see Figure 3.2):

Figure 4.2: Plots of maximum of magnitude of velocity ($|u|_{max}$) versus time: (a) the capillary waves at the early times and (b) the spurious currents and the effect of $C_{pc}$ in eliminating them.

In this section we focus on analysing the sensitivity of the results on the magnitude of compression coefficient for capillary pressure. We use the original CSF formulation as well as the SSF with three different compression coefficients $C_{pc} = 0.3, 0.45$ and 0.49. The results for maximum of the magnitude of velocity of all grid blocks are shown in Figure 4.3. These results are reported for a mesh of size $20^3$ and hence the droplet has a volume of $8^3$ grid blocks. As it is clear from these results the values of spurious currents are unacceptable for the
original CSF formulation in the presence of high surface tension forces. Therefore, the CSF formulation, in its original form, is not considered suitable for modelling multiphase flow when the capillary number is low. Our simulations show that a value of \( C_{pc} = 0.49 \) is effective in eliminating spurious currents in static problems. A side-by-side comparison of the CSF and SSF results is given in Figure 4.3.

![Figure 4.3: A side-by-side comparison of a cross section of the 3D droplet using 40^3 grid blocks, for CSF (left half) and SSF (right half) formulations. The cross sectional area is coloured by \( p_c \) and the cones show the spurious velocities at time \( t = 0.0002 \), when the spurious currents start to grow.](image)

The numerical results presented in Figure 4.2 show that the problem of violating the zero net capillary force constraint is not critical in solving for static problems using the sharp surface force model, as the velocities converge to zero. The reason for this is considered to be the tendency of the numerical scheme to align the interface to the mesh. In other words, the droplet tends to adjust itself to the mesh and find an equilibrium position. Nevertheless, this stable and spurious-free solution for static problems is necessary in order to evaluate the accuracy of the numerical method in calculation of curvature and consequently capillary pressure. In dynamic situations the constraint of zero net capillary force is not obeyed neither in SSF formulation nor in CSF formulation; this problem will be studied separately in section 4.3.

### 4.1.2 Effect of mesh resolution

The simulation results for modelling a static droplet are given in Table 1, for different mesh resolutions and different values of compression coefficients for \( \alpha \) and \( p_c \) (\( C_\alpha \) and \( C_{pc} \) respectively).

<table>
<thead>
<tr>
<th>( C_{pc} )</th>
<th>( C_\alpha )</th>
<th>( P_c ) relative error</th>
<th>( \mathbf{uu}_{max} ) (m/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10 \times 10 \times 10 )</td>
<td>( 20 \times 20 \times 20 )</td>
<td>( 40 \times 40 \times 40 )</td>
<td>( 10 \times 10 \times 10 )</td>
</tr>
<tr>
<td>0.3</td>
<td>1</td>
<td>0.08</td>
<td>0.04</td>
</tr>
<tr>
<td>0.45</td>
<td>1</td>
<td>0.1</td>
<td>0.06</td>
</tr>
<tr>
<td>0.49</td>
<td>0</td>
<td>0.11</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.11</td>
<td>0.05</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0.11</td>
<td>0.06</td>
</tr>
<tr>
<td>1.5</td>
<td>0.11</td>
<td>0.08</td>
<td>0.06</td>
</tr>
</tbody>
</table>

Table 1: Effect of mesh resolution, \( C_\alpha \) and \( C_{pc} \) on the convergence of velocity and capillary pressure for the static droplet.

The data in Table 1 are average values over a time interval of 0.00045-0.0005 seconds. These results show that if the capillary pressure becomes sharp enough (i.e. \( C_{pc} \rightarrow 0.5 \)) the spurious currents will be eliminated. Moreover, they show the convergence of calculated capillary pressure to the theoretical value of \( p_c = 5642 \) with mesh refinement.

### 4.1.3 Effect of smoothing indicator function

One approach for increasing the accuracy in calculating the interface curvature is to use a smoothed indicator function for calculation of the interface normals. Different smoothing kernels (SK) can be used for this purpose. A simple approach is to interpolate the indicator function from cell centres to the cell faces and then averaging...
the values of the cell faces to obtain a smoother representation of $\alpha$ at cell centres, which can be repeated recursively for $i$ times as:

$$\alpha_{\text{smooth}, i+1} = \langle \langle \alpha_{\text{smooth}, i}^f \rangle_{c-f} \rangle_{f-o}, \quad \alpha_{\text{smooth}, 0} = \alpha. \quad (4.2)$$

The effect of applying this smoothing kernel on $\alpha$ for zero, one and two times recursively is presented in Table 2. These results show that using a smoother $\alpha$ will increase the accuracy of capillary pressure. Preliminary studies, however, show that when calculating the curvature from a smoothed $\alpha$, the simulation results are not spurious free or the droplet is not kept stationary in some test cases, especially when using fine mesh files. In addition, in dynamic problems, destabilization of the interface was observed when using the smoothing method discussed above, where $\alpha$ and $\alpha_{\text{smooth}}$ were decoupled. This shows that more stable approaches for smoothing $\alpha$ is necessary to achieve higher accuracy in calculating capillary pressure and at the same time to avoid the destabilizations due to the decoupling of $\alpha$ and $\alpha_{\text{smooth}}$.

| Smoothing Kernel (SK) | $P_c$ relative error | $|u|_{\text{max}}$ (m/s) |
|-----------------------|----------------------|------------------------|
| 0                     | 10x10x10             | 20x20x20               | 40x40x40               | 50x50x50               | 10x10x10             | 20x20x20               | 40x40x40               | 50x50x50               |
| 0                     | 0.1058               | 0.0590                 | 0.0569                 | 0.0204                 | 2.8E-6                | 1.8E-6                | 4.0E-6                | 2.0E-8                |
| 1                     | 0.1009               | 0.0427                 | 0.0368                 | 0.0114                 | 7.3E-6                | 4.9E-7                | 7.6E-6                | 1.4E-6                |
| 2                     | 0.0983               | 0.0358                 | 0.0250                 | 0.0078                 | 1.2E-5                | 1.3E-6                | 2.9E-5                | 5.3E-5                |

Table 2: Effect of smoothing $\alpha$ for calculation of curvature. Smoothing kernels: 0: no smoothing, 1: one step smoothing (eq. (4.2) with $i = 1$), 2: 2 step smoothing (eq. (4.2) with $i = 2$).

4.1.4 Effect of time-step size

In this study the surface tension forces are applied in the numerical method as an explicit term. This will impose a limit on the time step by introducing a numerical capillary time scale; the numerical method is stable when the time step resolves the propagation of capillary waves (Brackbill et al., 1992):

$$\delta t < \left[ \frac{\langle \rho \rangle \delta x^3}{2 \pi \sigma} \right]^{1/2}, \quad (4.3)$$

where $\delta x$ is the size of grid blocks and $\langle \rho \rangle$, is the average density at the interface.

For the test case discussed in this section this imposes a limit of $\delta t < 0.2 \mu s$ and $\delta t < 0.07 \mu s$ for the mesh resolutions of $20^3$ and $40^3$ respectively. If the time-step size is not small enough, the capillary waves will not disappear. Table 3 shows this behaviour, represented in terms of the maximum of magnitude of velocity field averaged over a time interval of $t = 0.9 - 1 ms$.

| $dt$ (\mu s) | $|u|_{\text{ave}}$ (m/s) |
|--------------|-------------------------|
| 20x20x20     |                         |
| 40x40x40     |                         |
| 0.1          | 1.3E-11                 | 2.9E-10                 |
| 0.5          | 7.1E-10                 | 1.3E-9                  |
| 1            | 9.1E-6                  | 0.44                    |
| 2            | 0.4                     | 0.41                    |

Table 3: Effect time step size on vanishing capillary waves.

Although these results show that the time step size can be slightly (by a factor of about 5) higher than what is predicted by equation (4.3), the time step size can be the main limitation in the performance of simulations when modelling multiphase flow at low capillary numbers on image of porous rocks with a typical size ($: mn$).

4.1.5 Closure

The simulation results for this test case show that the capillary pressure converges to the theoretical solution by using mesh refinement, by controlling the sharpness of capillary pressure and by smoothing the indicator function for calculation of interface curvature, when the time step size does not exceed the capillary time limit. Moreover, it was observed that the spurious currents problem can be solved by using a sharp surface force model. The violation of zero net capillary force was not observed in this static problem, in the case of using SSSF model. In dynamic situations, when the capillary number is low, the accuracy of the simulation results can be significantly affected by the violation of zero net capillary force. The errors in applying the contact angle in modelling fluid-rock interactions are even more destructive. The following sections discuss these defects in the numerical method and the solutions we use to resolve them.
4.2 Droplet in contact with a flat plate

Solid wall adhesion effects play an important role in multiphase flow in porous media. In this section we present the numerical results for modelling a droplet resting on a flat plate with different contact angles. The theoretical solution for the equilibrium shape of the droplet is a spherical cap with radius:

\[
R = \left( \frac{3 V_\alpha}{\pi (1 - \cos \theta)^2 (2 + \cos \theta)} \right)^{1/3}
\]

where \( V_\alpha \) is the volume of the droplet calculated by numerically integrating \( \alpha \) over the entire flow domain. Consequently the theoretical value of capillary pressure will be:

\[
P_{c, \text{Theoretical}} = \frac{2 \sigma}{R}
\]

We use different representations of geometry of the system in a uniform Cartesian mesh to study the effect of alignment of mesh to the solid surface. The system studied is the static solution for a droplet of liquid on flat plates described mathematically by equation \( y = a_w x + c_w z \). The solution domain is limited to a hemisphere. The liquid drop is only in contact with the plate and not in touch with the hemisphere. The mesh file for a grid resolution of \( \delta x = 5 \mu m \) is shown in Figure 4.4.

![Different representations of a flat solid wall in a uniform Cartesian mesh. Left: The wall is perpendicular to coordinate system, Right: The wall equation is \( y = 1.0x + 0z \), where \( y \), \( x \) and \( z \) are axes aligned to the mesh.](image)

4.2.1 Smooth solid walls

Our simulations show that the formulation given in equation (3.8) for interface normals produce large errors in the calculated capillary pressure when the contact angle is low. To alleviate this problem interface normals for those faces in the mesh file that have a common edge with the solid boundaries are modified. The interface normals for these faces are obtained by averaging the value of the interface normals that are obtained from the gradient of the indicator function (eq. (3.6)) and the interface normals that are obtained from contact angle model (eq. (0.3)).

The results of simulations for the flat plate with equation \( y = 0 \) are summarised in Table 4 in terms of the calculated capillary pressure and magnitude of maximum velocity for four different contact angles.

| \( N_x \) | \( \theta \) | \( P_c \) | \( P_c \) relative error | \( |u|_{\text{max}} \) |
|---|---|---|---|---|
| 20 | 30 | 761.545 | 0.101 | 2.1E-03 |
| 60 | 1626.140 | 0.023 | 3.5E-04 |
| 90 | 2419.540 | 0.033 | 3.7E-05 |
| 150 | 3038.110 | 0.034 | 4.4E-03 |
| 40 | 1596.940 | 0.007 | 7.3E-03 |

Table 4: Summary of the results for a droplet resting on a flat plate (with equation \( y = 0 \)), simulation parameters: \( C_{pc} = 0.49 \) and \( SK=0 \) with different contact angles.

4.2.2 Jagged solid walls

The simulation results for different representation of a flat surface in a Cartezian mesh (i.e. different values of \( a_w \) and \( c_w \)) are shown in Table 5. These results show that the simulations produce spurious velocities when the interface is in touch with a wall that is not aligned with the mesh, which is referred to as jagged walls in this
report. The reason for this behaviour can be inaccurate evaluation of curvature as well as bad approximation of the interface Dirac-delta function (see eq. (3.11)) close to jagged solid walls.

Table 5: The simulation results for different values of \( a_w \) and \( c_w \), showing the non-physical velocities generated close to the jagged solid walls. The results are averaged over a time interval of \( t = 0.9-1 \text{ ms} \). Other simulation parameters are: \( C_p = 0.49 \), \( \theta = 90^\circ \), \( C_a = 1 \), \( \sigma = 0.05\text{N/m}^2 \).

Visualization of the velocity field in these cases shows that the non-physical velocities generated close to the jagged solid walls are parallel to interface. To deal with this problem we decided to filter those components of capillary pressure fluxes which are parallel to interface. This filtering is achieved by using the following operator to correct the capillary force \( f_{c,f} \):

\[
f_{c,f,\text{filtered}} = f_{c,f} - f_{c,f,\text{filtered}},
\]

where \( f_{c,f,\text{filtered}} \) is a time dependent term, defined at face centres. Starting from a zero as initial condition, it is updated as follows:

\[
f_{c,f,\text{filtered}} = \frac{\delta f_f}{(|\delta_f| + \varepsilon)} \left( f_{c,f,\text{old}} + C_{f \text{,filtered}} \left( \nabla p_c - (\nabla p_c \cdot \mathbf{n}_f) \mathbf{n}_f \right) \mathbf{n}_f \right),
\]

and \( f_{c,f,\text{old}} \) represent the values of \( f_{c,f,\text{filtered}} \) at previous time-step. The term \( \frac{\delta f_f}{(|\delta_f| + \varepsilon)} \) restricts this correction term to the interface, which gradually dampen those component of \( \nabla p_c \) that are parallel to interface, so that they finally go to zero. This correction may introduce small errors in the calculations, however the effect of this filtering was observed to be negligible in the absence of non-smooth solid boundaries. \( C_{f \text{,filtered}} \) is a coefficient determining how fast the non-physical velocities are filtered; a value of 0.005 was found to be sufficient in most of our simulations. The simulation results after filtering velocities parallel to interface are presented in Table 6.

Table 6: Simulation results for modelling the droplet resting on different flat surfaces with equation \( y = a_w x + c_w z \), where \( y, x \) and \( z \) are axes aligned to the mesh. The results are for a mesh size of \( 20^3 \) (\( \delta x = 2.5 \text{ \mu m} \)). Other simulation parameters are: \( C_p = 0.49 \), \( \theta = 60^\circ \), \( C_a = 1 \).
4.2.3 Closure

The results for predicted static capillary pressure presented for this test case show that the accuracy of calculations for static contact angle is acceptable. The results for the velocity show that in the case that the solid walls are jagged, non-physical velocities are generated parallel to the interface. This problem is solved by filtering the capillary fluxes generated parallel to interface at the vicinity of jagged solid walls. The effects of different approaches for applying the wall adhesion effects in dynamic situations will be presented in section 4.4.

4.3 Moving droplet

As it was stated earlier, the constraint of zero net capillary force is important when the magnitude of capillary forces are high compared to the viscous forces. Our simulations show that both CSF formulation and SSF formulation suffer from the violation of this constraint when the capillary number is low.

In the CSF formulation the results are affected by both the spurious currents and the violation of zero net capillary force constraint. In SSF formulation the spurious velocities are not present, but the calculations are affected by the violation of zero net capillary force, where this non-zero net capillary force tends to attach the droplet to the mesh and prevents its movement.

To deal with this problem we decided to filter a fraction of the capillary fluxes \( \phi = S f \left( f_{c,f} - \nabla \cdot \nabla p_c \right) \). This filtering will explicitly set the capillary fluxes to zero when their magnitude is in the order of the numerical errors. The filtered capillary flux then reads:

\[
\phi_{c,\text{filtered}} = \phi_c - \min\left[ 1, C_{\text{fc, filt}} f_{c,f} \right] \frac{\phi_c}{1 + \varepsilon},
\]

where \( l f_{c,f} \) is the maximum of the discretised capillary force (see equation (3.10)) over the entire solution domain.

Moreover, we observed that using a weighted interpolation method for obtaining the face centred curvature from the cell centres alleviates the problem of violation of zero net capillary force constraint. The weighted interpolation method used for this purpose is as follows:

\[
f_x = \left( f_x k \right) + \left( f_a \alpha \right), \quad f_a = \alpha (1 - \alpha) + 0.001
\]

where the cell to face interpolations, \( f_x k \) and \( f_a \) are obtained using the linear interpolation scheme given in equation (0.14).

In this section, we study the effect of the filtering technique discussed above in enforcing the zero net capillary force constraint by modelling steady movement of a micro-scale droplet. The case selected for this purpose is the steady movement of a droplet with a radius of \( R = 20 \mu m \), suspended in another liquid moving with a uniform velocity. The fluid properties are the same as static droplet case presented in section 4.1. The solution domain is constrained to a cylinder in a stationary Cartesian mesh, in order to speed up the calculations. Zero-gradient boundary condition is used for all physical variables, except for the velocity on the left side where a uniform fixed-value (Dirichlet) boundary condition is used. A visualization of a cross section of the solution domain is shown in Figure 4.5. This figure shows the destructive effect of the numerical errors in calculation of surface tension forces, and the effect of filtering method discussed above in alleviating this problem.

![Figure 4.5: A visualization of the velocity field for the droplet moving in a uniform velocity field of \( u = 0.001 \text{ m/s} \) in a Cartesian mesh with resolution of \( \delta x = 2 \mu m \), using (a) CSF: \( C_{pc} = 0 \), (b) SSF: \( C_{pc} = 0.49 \) formulations and (c) an intermediate formulation with \( C_{pc} = 0.20 \) where the capillary fluxes are filtered using equation (4.8) to enforce the zero net capillary force constraint.](image-url)
The solution to this problem is that the droplet should remain spherical and move with a constant velocity. Table 7 summarises the results for this case using different values of $C_{\phi,\text{filt}}$ and $C_{pc}$. These results are average values over a time period in which the droplet moves a distance of $2R = 40 \, \mu m$.

<table>
<thead>
<tr>
<th>$C_{pc}$</th>
<th>$N_c$</th>
<th>$N$</th>
<th>SK</th>
<th>$C_{pc,\text{filt}}$</th>
<th>$C_{\phi,\text{filt}}$</th>
<th>$Pc$ error</th>
<th>$u$ relative deviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.2</td>
<td>2.0E-04</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.000</td>
<td>2.23%</td>
<td>2.18%</td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-05</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.000</td>
<td>10.61%</td>
<td>14.07%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-05</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.000</td>
<td>3.57%</td>
<td>-1.28%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-05</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.000</td>
<td>4.75%</td>
<td>-84.61%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-05</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.000</td>
<td>1.97%</td>
<td>90.92%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-05</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>0.64%</td>
<td>-32.39%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>3.51%</td>
<td>1.22%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>5.20%</td>
<td>12.22%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>4.02%</td>
<td>0.64%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>4.23%</td>
<td>3.26%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>4.21%</td>
<td>-0.17%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>4.03%</td>
<td>-0.06%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>5.39%</td>
<td>0.92%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>4.77%</td>
<td>0.21%</td>
<td></td>
</tr>
<tr>
<td>0.49</td>
<td>2.0E-06</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0.005</td>
<td>2.68%</td>
<td>2.72%</td>
<td></td>
</tr>
</tbody>
</table>

Table 7: Summary of the simulation results for a droplet moving with a constant velocity. “$u$ relative deviation” is the relative deviation of the droplet velocity from its theoretical value.

Our simulations show that a value of $C_{pc} = 0.20$ and a threshold of $C_{\phi,\text{filt}} = 0.005$ for filtering capillary fluxes is a good selection for dynamic multiphase flow problems at micro-scales. In addition, these results show that when the interface curvature is obtained from a smoothed indicator function, the problem of violating the constraint of zero net capillary force is alleviated and less filtering of capillary fluxes is required. In addition, the filtering method presented in equations (4.6) and (4.7) with a coefficient of $C_{\phi,\text{filt}} = 0.005$, which is used to filter the capillary fluxes parallel to interface at interface region, was found to be effective in eliminating the spurious currents.

4.4 Capillary tube

The dynamics of contact angle is one of the most important parameters controlling flow behaviour in porous media. Despite its simplicity, however, it remains only partially understood (Dupont and Legendre, 2010, and references therein). In this section, we study the behaviour of contact angle dynamics using numerical experiments on two-phase flow in capillary tubes. We will use this test case to validate the numerical model as well as to tune the simulation parameters for multiphase flow simulations on more complex rock images. Similar to the case of modelling a droplet resting on a flat plate, we consider different voxelized representation of the geometry. A visualization of two sample mesh files used for this purpose is given in Figure 4.6.
The equation for cylinder axis is $y = a_w x + y_0$, $z = c_w x + z_0$ where $a_w = c_w = 0$ for the left tube and $a_w = 0.5$ and $c_w = 0$ for the right tube. The mesh size is $N_e = 20$.

4.4.1 Smoothing solid wall normal vectors

When solid walls are not aligned with the Cartesian mesh (i.e. $a_w \neq 0$ or $c_w \neq 0$), a question arises on how to define the normal vector to the walls, $n_w$, used in equation (0.3) for calculating the normal vector to the interface at solid boundaries, $n_{|n}$. The first approach is to use the normal vector of the faces on the solid walls as the wall normal vector; $n_{w |n}$. Our simulations show that this approach produces large errors in dynamic situations in modelling contact line movement. Another approach is to smooth the solid-wall normals using appropriate smoothing kernels. We use a similar approach as equation (4.2), by replacing $\alpha$ by $n_w$.

$$n_{w,\text{smooth},i} = n_{w |n} \frac{\langle \langle n_f |_{w} \rangle_{c \rightarrow p} \rangle_{p \rightarrow n}}{\langle \langle n_f |_{w} \rangle_{c \rightarrow p} \rangle_{p \rightarrow n}}, \quad n_{\text{smooth},b} = n_{f |w}, \quad i = 0,1,2,...$$  \hspace{1cm} (4.10)

where $\langle \langle \rangle_{c \rightarrow p}$ stands for interpolating a face centred field to obtain a point centred field, defined at the corners of grid blocks, and $\langle \langle \rangle_{p \rightarrow n}$ means interpolating back the point-centred field to obtain a face-centred field. The numerical results for wall smoothing kernels (WSK) of 0, 1 and 3 (i.e. eq. (4.6) with $i = 0$, 1 and 3, respectively) are given in the following section in Table 8.

4.4.2 Low capillary numbers

In order to make the analysis of results simple, in this section we consider the steady displacement of one liquid by another liquid in a capillary tube of diameter $25 \mu m$. We apply different dynamic pressure gradients along the tube to be able to model the two-phase flow at different capillary numbers. In other words, the boundary conditions used for pressure is fixed-value for dynamic pressure for left and right sides of the tube and zero-gradient boundary condition for the velocity field on either sides of the tube. For simplicity, in this section we only analyse the velocity field when it reaches its maximum value over time, in the absence of gravity. The average fluid velocity for this problem can be obtained by equating the viscous damping force to the net driving force due to dynamic pressure gradient:

$$|u_{\text{avg}}| = \frac{1}{8\mu} \frac{\Delta p_d}{L} R^2,$$  \hspace{1cm} (4.11)

where $L$ is the length of capillary tube and $R$ is its radius.

Our simulations show that the simulation results are affected by the violation of zero net capillary force constraint for low capillary numbers. The simulation results for different representation of the geometry, and different selection of simulation parameters are summarised in Table 8.
Table 8: Simulation results for modelling steady two-phase flow at a low capillary numbers. “$|u|_{\text{rel. error}}$” is deviation of average velocity from the value of $|u_{\text{avg}}|$ predicted by equation (4.11). “$|u|_{\text{max. local error}}$” is the relative deviation of maximum of velocities of all grid blocks from the predicted maximum velocity using equation (4.11), averaged over entire time domain.

The simulation results presented in this table show the importance of filtering capillary fluxes, in order to eliminate the numerical errors introduced in the velocity field as a result of inaccurate computation of capillary forces. Since the numerical errors in the capillary forces are higher than the viscous forces at very low capillary numbers, the simulations becomes instable if the capillary fluxes are not filtered. The filtering method given in equation (4.7) is effective in alleviating this problem. However, the results presented in Table 8 show large deviations of the predicted velocity field for some of the mesh configurations, even when the capillary fluxes are filtered. A comparable magnitude of fluctuations is expected in real conditions as a results of contact angle hysteresis effects; evidence of which can be observed in experimental data in literature such as Schiffer and Wong (2000), for rough surfaces.

4.4.3 High capillary numbers
At high capillary numbers, when using no-slip boundary condition for velocity, the average velocity of contact line movement is underpredicted when it is compared to the equation (4.11). In these situations the effect of spurious velocities and the errors in surface tension forces are less compared to the low capillary number flow regime. The results for different capillary numbers is given in Table 9.
These results show that a slip boundary condition for velocity field should be used in the contact line region so that the numerical results much the theoretical solutions and experimental observations. Our primarily simulations with a slip boundary condition for contact line shows that the slip boundary condition is effective in eliminating the deviation of predicted contact line velocity from the theoretical solution. Studying the required magnitude of slip velocity for different mesh resolutions, capillary numbers and different representation of solid walls in the cartesian mesh is considered in our future work.

4.4.4 Closure

The numerical results presented in this section show that the errors introduced because of inaccurate implementation of contact angle can affect the accuracy of the results at low capillary numbers. Specifically, existence of jagged solid walls will result in high spurious velocities parallel to the interface. We showed that this problem can be alleviated by filtering the capillary fluxes generated parallel to the interface.

In the numerical results presented in this section, a single contact angle was used in the numerical model. The effect of contact angle hysteresis will be studied in future work. We will improve our numerical method to be able to use the full set of dynamic contact angle models available in literature. At the moment, our code can support the dynamic contact angle as a function of shear rate as an input parameter. Including the effects of stick-slip behaviour of contact line, which is common in real physical phenomena, will be considered in future work. Specifically we will model the same problems as the experiments in Schaffer and Wong (2000), as their experiments covers the range of capillary numbers occurring in porous media.

| $N_{ca}$ | $N$ | $P_c$ | relative error | $|\mathbf{u}|_{avg}$ | $|\mathbf{u}|$ rel. deviaton | $|\mathbf{u}|$ max. deviaton |
|--------|----|------|----------------|----------------|----------------------|----------------------|
| 0.013  | 10 | 3642 | 0.0612         | 0.63151        | -0.21                | 0.00304              |
| 0.002  | 10 | 3625 | 0.0562         | 0.119882       | -0.25                | -0.0051              |
| 3E-04  | 10 | 3591 | 0.0462         | 0.013219       | -0.17                | 0.04695              |
| 1E-04  | 10 | 3584 | 0.0442         | 0.007064       | -0.11                | 0.40056              |
| 7E-05  | 10 | 3585 | 0.0444         | 0.003695       | -0.07                | 0.46102              |
| 7E-05  | 40 | 3624 | 0.0491         | 0.003358       | -0.15                | 0.97928              |

Table 9: Simulation results for modelling steady two-phase flow at high capillary numbers.
5 Future work

5.1 The numerical method

The numerical method discussed in this report needs to be improved in different aspects:

1  **Contact line slip velocity model:** The state of stresses in the vicinity of contact line can not be described using the conventional Navier-Stokes equations, as the momentum balance equation with standard no slip boundary condition produce an infinite viscous dissipation; see Dupont and Legendre (2010) and references therein. This divergence of strain rate is responsible for a slip velocity which pulls contact line as the interface moves. At the moment our numerical method introduce a slip velocity which depends on the size of grid blocks. The magnitude of this slip velocity shall be tuned so that the numerical method converges to the correct physical solution for the contact line movement.

2  **Contact angle hysteresis models:** Implementing different contact angle hysteresis models available in literature is considered in our future work. Appropriate implementation of the contact angle hysteresis effects are important in pore-scale modelling of multiphase flow in porous media, where rock surfaces are both topologically and chemically heterogeneous leading to significant hysteresis effects.

3  **Computational efficiency:** The efficiency of the numerical method shall be improved by alleviating the requirement for the capillary time step limit for low capillary number flow regimes. Implementing the capillary forces in a semi-implicit manner (i.e. by solving the equations for advection of indicator function (eq. 3.4) and the capillary pressure equation (eq. 3.13) inside the PISO loop) is considered as a potential solution for alleviating the capillary time limit which increases the simulation time drastically for low capillary number flow regimes, as it will be discussed later in this chapter.

5.2 Pore-scale methodology

The numerical method discussed in this report gives us the opportunity to study two-phase problems occurring in different geological processes. Specifically we will focus on calculating the relative permeability and capillary pressure curves in two important geological processes:

1  **CO₂ storage applications:** studying the the injection of CO₂ into aquifers.

2  **Enhanced Oil Recovery (EOR) applications:** The two-phase flow problems related to EOR processes, such as the drainage process in displacement of oil by water in different types of porous media; sandstones, carbonates and sand packs.

5.3 A preliminary case study on micro-CT images of porous rocks

In this section we perform two-phase flow calculations on a micro-CT image of a sandstone of size of $\phi N^3 = 0.2 \times 100^3 \mu m$, referred to as Berea100 in this report. The purpose is to demonstrate the capabilities of our numerical approach, and to describe an example of what we will focus in our future work. A visualization of the mesh and the fluid interfaces at time $t = 0.03 s$ for a drainage experiment is shown in Figure 5.1.

![Figure 5.1](image)

**Figure 5.1.** A visualization of the solution domain for Berea100 sandstone. Oil is coloured in dark red and water in light green. The arrows on the right side show the velocities generated as a result of the applied dynamic pressure difference on the image.
As it was stated earlier in Section 1.1, many different flow patterns occur in multiphase flow through porous media in geological processes. In this section we only consider two scenarios for our numerical experiments. First, we model the displacement of a wetting phase (water), which is initially filled the whole solution domain, by a non-wetting phase (oil). This experiment is referred to as a primary drainage experiment. Next, the final fluid distribution ($\alpha$ field) in the porous rock obtained from the primary drainage experiment is used as the initial condition for a new numerical experiment, referred to as an imbibition experiment, where water, the wetting phase, is injected.

The capillary number used in these experiments is of the order of $N_c = \mu u_d / \sigma = 2 \times 10^{-4}$, where $u_d$ is the Darcy velocity in the $x$ direction. The contact angle model used in this experiments is a fixed value of $\theta = 30^\circ$, $\sigma = 0.01 \text{Nm}^{-1}$ and other properties of the fluids are the same as water at room temperature.

In the primary drainage experiment the boundary conditions used for dynamic pressure are a fixed-value of $p_{\text{left}} = 1000 \text{ Pa}$ and $p_{\text{right}} = 0 \text{ Pa}$ for the left and the right sides of the image, respectively. A zero-gradient boundary condition is used for the velocity field. The boundary condition for the indicator function is zero-gradient for the right patch and fixed-value of $\alpha = \alpha_w = 1$ (100% oil) for the left patch. The same boundary conditions are used for the imbibition experiment except for the indicator function on the left patch where a fixed-value of $\alpha = 0$ ($\alpha_o = 1$, 100% water) is used.

We run the simulations for a total time of $0.1 \text{ s}$. The time step size was controlled by a Courant-Friedrichs-Lewy (CFL) condition of (Courant et al., 1928): $|u| \delta t / \delta x \leq 0.4$ which limits the time step to be roughly $3 \times 10^{-6} \text{ s}$. The total simulation time was 30 hours on a single 3.3 GHz processor.

The results are reported in Figure 5.2 in terms of the fractional flow curves. These results are obtained by averaging the values of relative fluxes (the total flux of each phase divided by the total flux of all phases, $q_a/(q_a + q_w)$, $a = o, w$) obtained on several cross sections (slices) along the image and over the last 90% of the simulation time. For this purpose a number of 50 slices were used normal to the $x$ direction, and every $0.001 \text{s}$ during a total time interval of $0.09 \text{s}$.

![Figure 5.2: Fractional flow curves obtained from the numerical primary drainage and imbibition experiment.](image)

Relative permeability and capillary pressure curves are not calculated at this stage. Usually these curves are obtained from the steady-state laboratory experiments, in which the saturations and total fluxes remain constant over time. Comparing the results obtained from transient experiments with the steady-state experiments (both laboratory and numerical) is considered in our future work.
5.4 Simulation time

The simulation time of linear equation solvers at each time step increases at least linearly by increasing the number of cells \( N_{\text{cells}} = \phi N^3 \), where \( N \) is the number of voxels at each direction and \( \phi \) is the porosity of rock. The total simulation time is then proportional to \( t_{\text{sim}} \propto \phi N^3 N_t = \phi N^3 \Delta t / \delta t \).

In simulations on porous rock images, it is desired to continue simulations until a specified amount of the fluid \( V_{\text{inj}} \) in the pores is displaced by the injecting fluid: \( \Delta t = \frac{V_{\text{inj}}}{A_d u_d} \propto \frac{N \delta t}{u_d} \), where \( A_d \) is the cross sectional area of the porous rock.

At high capillary numbers the time step size, \( \delta t \), is limited by the CFL condition: \( \delta t \propto \delta t / u_d \). Therefore the simulation time scales as \( t_{\text{sim}} \propto \phi N^4 \). Consequently, if we want to perform the simulations on a mesh of size 2 times bigger in each direction or if we want to refine the mesh by a factor of 2 in order to achieve a higher accuracy, the simulation time will be \( 2^4 \) times higher.

At low capillary numbers the situation is worse as the time step size is limited by the capillary time limit (eq. (48)): \( \delta t \propto \left( \frac{\rho}{\sigma} \right) \delta t^3 / \sigma \). Therefore the simulation time scales as \( t_{\text{sim}} \propto \phi N^{4.5} \text{Re}^{-1/2} N_c^{-1/2} \). In other words, the simulation time is inversely proportional to the Darcy velocity used in the simulations, when considering all other parameters constant. Therefore using the approach used in this study it is difficult to do the numerical simulations at very low capillary numbers (roughly speaking \( N_c < 10^{-7} \)) on large image files (\( N > 100 \)). This limitation can be overcome by developing a more implicit implementation of surface tension forces in the Navier-Stokes equations in order to allow larger time steps, or by using parallel computing and GPU technology in reducing the simulation time. Multiphase flow simulations at low capillary numbers can be done efficiently using the pore-network models that produce satisfactory results for the low capillary number flow regimes.

5.5 Closure

To be able to study these processes, however, appropriate up-scaling techniques to be able to extract the macroscopic properties from the minimum calculations, hence less computational cost, of the direct numerical simulation of multiphase flow at the pore scale, DNSMFPs, is important in reducing the computational costs. Reducing the computational costs is, in turn, important so that DNSMFPs can be used as a practical substitution/supplement to the time-consuming and expensive experiments that are nowadays used for measuring macroscopic properties of porous rocks such as relative permeability and capillary pressure curves.

In addition micro-CT technique allows imaging of porous rocks saturated with different fluids (see for example Iglauer et al. (2010)). This allows us to compare the results of the numerical simulations and experimental data in a voxel-by-voxel basis and is considered as a future work.
Imperial College Consortium on Pore-Scale Modelling

References


Shyy, W. Computational fluid dynamics with moving boundaries. Taylor & Francis, 1996.


Appendices

Finite volume method

In general, the numerical solution of differential equation is obtained by:

**Temporal and spatial discretisation:** the spatial domain is subdivided into a finite number of smaller volumes. The temporal space is also subdivided into small intervals or time-steps.

**Equation discretisation:** All the variables in the continuous space are replaced with a list of collocated variables, i.e. at discrete points in time and space. The differential operators in the continuous domain are replaced with corresponding discretised operators, which act on the collocated variables. The differential equations are converted into a set of discrete equations by replacing the differential operators with corresponding discrete differential operators and the values of the corresponding continuum variable be as close as possible. Different criteria for the analysis of this conversion of PDEs to SAEs are developed during the past decades, including consistency, stability and accuracy. Discussion of these analyses can be found in the literature (Jasak, 1996). The numerical solution will be obtained by solving the set of discrete equations, which usually is achieved by linearising these equations and solving the linear equation using appropriate algorithms.

In this Appendix we will discuss some of the well known discretisation schemes which are used in discretisation of the PDEs discussed in this report. Some of the material presented here are taken from OpenFOAM (2009b).

**Temporal discretisation**

If we denote all the spatial terms as \( A_x(\phi) \) where \( A_x \) represents any spatial operator, e.g. Laplacian, then we can express a transient PDE as:

\[
\partial_t (\rho \phi) + \sum A_x(\phi) = 0
\]  

(0.1)

In the finite volume method, for a particular cell, \( i \), we take the volume integral over the total volume of the cell \( (V_i) \), and time interval \([t, t + \Delta t]\), the resultant integral form will be:

\[
\frac{1}{V_i} \int_{t}^{t + \Delta t} \rho \phi d\Omega_v + \frac{1}{V_i} \sum A_x(\phi) dV = 0
\]  

(0.2)

The first term can be expressed as

\[
\frac{1}{V_i} \int_{t}^{t + \Delta t} \frac{\partial}{\partial t} \rho \phi d\Omega_v = \frac{1}{V_i} \left( \rho \phi V_i \right)_{t + \Delta t} - \left( \rho \phi V_i \right)_t
\]  

(0.3)

In the case that continuum variables are collocated at either ends of time interval discretisation of this term can be expressed as:

\[
\partial_t (\rho \phi) = \frac{(\rho \phi)^n - (\rho \phi)^{n-1}}{\Delta t}
\]  

(0.4)

The second term in equation (0.2) can be expressed as

\[
\frac{1}{V_i} \int_{t}^{t + \Delta t} A_x(\phi) d\Omega_v dt = \frac{1}{\Delta t} \int_{t}^{t + \Delta t} A_x^* \phi dt
\]  

(0.5)

where \( A_x^* \) represents the spatial discretisation of \( \frac{1}{V_i} \int A_x d\Omega_v \). The discretisation of the time integral can be obtained in three ways:

**Euler implicit** uses implicit discretisation of the spatial terms, thereby taking current values \( \phi^n \).

\[
A_x(\phi) = A_x^*(\phi)^n
\]  

(0.6)

In this case the contribution of \( A_x(\phi) \) on final matrix of equations will be a “matrix of coefficients”, [A], and the contribution to the source vector, \([b]_n\), will be zero.

**Explicit** uses explicit discretisation of the spatial terms, thereby taking old values \( \phi^{n-1} \).
In this case the contribution of \( A_x^* (\varphi) \) on final matrix of equations will be a “source vector”, \([b]_x\), and the contribution on matrix of coefficients, \([A]_x\), will be zero. This scheme is first order accurate in time and is unstable if the Courant number \((Co)\) (Courant et al., 1928): is greater than 1. The Courant number is defined as
\[
Co = \frac{u_{\text{avg}} \cdot d}{d^2 \Delta t}
\]  

Crank Nicholson uses the trapezoid rule to discretise the spatial terms, thereby taking a mean of current values \( \varphi^n \) and old values \( \varphi^{n-1} \).

\[
A_x^* (\varphi) = \frac{1}{2} \left( A_x^* (\varphi^n) + A_x^* (\varphi^{n-1}) \right)
\]

It is, however, possible that collocate the continuum variables to the centre of time interval, and therefore, the superscript \( n \) and \( n-1 \) will represent the collocated variables at times \( t + \Delta t/2 \) and \( t - \Delta t/2 \) respectively. In this case the formula (0.6) is the most suitable scheme for discretisation of \( A_x^* \), as the value \( \varphi^n \) are located at the centre of integration interval. In this case the discretisation of time derivative term can be obtained by simple differencing in time using:

**First-order backward differencing** scheme. This scheme is identical to the Euler implicit discretisation scheme, i.e. using equations (0.6) and (0.4), and it is first order accurate in time. This scheme will be referred to as Euler implicit in this report.

**Second-order backward differencing** scheme, that is second order accurate in time by storing the values from a time step previous to the last \( (\varphi^{n-2} = \varphi(t - 3\Delta t/2)) \):
\[
\partial_t (\rho \varphi) = \frac{3(\rho \varphi)^n - 4(\rho \varphi)^{n-1} + (\rho \varphi)^{n-2}}{2\Delta t}
\]

**Spatial discretisation**
In the finite volume method, the spacial domain is divided into control volumes. Each control volume is defined by the set of the faces \( (f) \) which bound the control volume. Each face in turn is defined by an ordered list of points which form the edges of the face. These points are used to calculate the properties of each face such as face outward normal vector \( (n_f) \) and area \( (|S_f|) \). (see Figure 3.1). The spacial discretisation is complete by replacing the primary variables with the collocated variables which are defined at the centre of the control volumes.

Most spatial derivative terms are then converted to integrals over the cell surface \( S_f \) bounding the volume using Gauss theorem:
\[
\int_{S_f} \nabla \sum\varphi dV = \int_{S_f} \sum n_f \varphi dS
\]
where \( n_f \) is the outward normal vector to the surface area bounding the cell volume \( V_i \). \( \varphi \) can represent any tensor variable continuous in space discrete in time \( (\varphi) \) is used to represent either \( \varphi^n \) or \( \varphi^{n-1} \) as used in (0.6) and (0.7) respectively). The star notation \( \sum \) in right hand side is used to represent any tensor product, i.e. inner (\( \cdot \)), outer and cross(\( \times \)), corresponding to the respective operator in the left hand side: divergence \( \cdot \nabla \), gradient \( \nabla \) and \( \nabla \times \).

The the integral form of PDEs, in general, will contain both surface and volume integrals. which will be discretised as follows:
\[
\int_{n_f \in S_f} n_f \sum \varphi_f dS = \sum_{S_f \in S_i} |S_f| n_f \sum \varphi_f = \sum_{S_f \in S_i} S_f \sum \varphi_f
\]

\[
\int_{V_i} \varphi dV = \varphi V_i
\]
The next step in obtaining the discretised equations is to define the values of $\varphi_f$ and in terms of collocated variables at cell centres. Different schemes can be used for this purpose. The following sections give an overview of these schemes for discretisation of the operators that will be used in the rest of this report. Treatment of boundary conditions can be found in Ubbink, (1997) and Rusche, (2002).

**Cell to face interpolation**

In this report we refer to the collocated variables defined at the centre of cells as volume field or simply fields. If the values of any field variable are needed in face centres, they are obtained by interpolation of the values of the field at adjacent cells. For the internal faces, the face field $\varphi_f$ can be evaluated using a variety of schemes:

**Central differencing (CD)** is second-order accurate but unbounded

$$\varphi_f = f_s \varphi_p + (1 - f_s) \varphi_N$$

(0.14)

where $f_s = fN/PN$ where $fN$ is the distance between $f$ and cell centre $N$ and $PN$ is the distance between cell centres $P$ and $N$.

**Upwind differencing (UD)** determines $\varphi_f$ from the direction of flow and is bounded at the expense of accuracy

$$\varphi_f = \begin{cases} 
\varphi_p & \text{for } F \geq 0 \\
\varphi_N & \text{for } F < 0 
\end{cases}$$

(0.15)

where $F$ determines the direction of flow and is defined in equation (0.20).

**Blended differencing (BD)** schemes combine UD and CD in an attempt to preserve boundedness with reasonable accuracy,

$$\varphi_f = (1 - \gamma)(\varphi_f)_{UD} + \gamma(\varphi_f)_{CD}$$

(0.16)

OpenFOAM has several implementations of the Gamma differencing scheme to select the blending coefficient $\gamma$. In addition it offers other well-known schemes such as van Leer, SUPERBEE, MINMOD etc. (OpenFOAM, 2009b). If the values of field variable are known, the results of the interpolation will be a new collocated variable defined at the centre of faces. In this case the collocated variable is called a face-field. In the case that an interpolation is required for a variable which is a function of primary variables, the results of interpolation, for each face, will be function of primary field variables. These functions will be linearised and used to form the matrix of coefficients.

**Laplacian operator**

The Laplacian term $(\nabla \cdot (\Gamma \nabla \varphi))$ is integrated over a control volume. Its discretised form can be obtained using equations (0.11) and (0.12) by replacing $\sum$ and $\varphi$ with $\cdot$ and $\Gamma \nabla \varphi$ respectively:

$$\nabla \cdot (\Gamma \nabla \varphi) = \sum_{f \in S_i} \Gamma_f \cdot (\nabla \varphi)_f$$

(0.17)

The values of $\Gamma_f$ will be interpolated from the face-centred values. The interpolation scheme will be linear unless stated otherwise. The face gradient discretisation $(\nabla \varphi)_f$, when the length vector $\mathbf{d}$ between the centre of the cell of interest $P$ and the centre of a neighbouring cell $N$ is orthogonal to the face plane (i.e. parallel to $\mathbf{S}_f$), is discretised as follows:

$$\mathbf{S}_f \cdot (\nabla \varphi)_f = |\mathbf{S}_f| (\nabla \varphi) = |\mathbf{S}_f| \frac{\varphi_N - \varphi_P}{|\mathbf{d}|}$$

(0.18)

Therefore, the Euler implicit discretisation of Laplacian term can be obtained by combining equations (0.6), (0.17) and (0.18):
\[
\n\nabla \cdot (\Gamma \nabla \varphi) = \sum_{j \in S_i} \Gamma_j |S_j| \frac{\varphi_{N_f}^j - \varphi_p^e}{d} \tag{0.19}
\]

where \( N_f \) stands for the neighbouring cell, sharing face \( f \) with cell \( P \).

**Convection term**

Similar to the Laplacian operator, the convection term is integrated over a control volume and linearised as follows:

\[
\int_{V_i} \nabla \cdot (\rho \mathbf{u} \varphi) dV = \sum_{j \in S_i} S_j \cdot (\rho \mathbf{u}), \varphi_f = \sum_{j \in S_i} F \varphi_f \tag{0.20}
\]

The face field \( \varphi_f \) can be evaluated using a variety of schemes as discussed in Section 0.

**Divergence operator**

The Divergence operator \( \nabla \cdot (\varphi) \) integrated over a control volume, and then applying Gauss theorem. Its discretised form can also be obtained using equations (0.11) and (0.12) by replacing \( \sum \) and \( \varphi \) with \( \cdot \) and \( \varphi \) respectively:

\[
\nabla \cdot (\varphi) = \sum_{j \in S_i} S_f \cdot (\varphi)_f \tag{0.21}
\]

The divergence term \( \nabla \cdot (\varphi) \) described in this section is strictly an explicit term that is distinguished from the convection term, i.e. in that it is not the divergence of the product of a velocity and dependent variable. It is mainly used to map a face-centred field to a cell-centred source field. In the case that the values of the field are defined in cell centres the face-centred values will be obtained by interpolation as discussed in section 0.

**Gradient operator**

The gradient operator \( \nabla (\varphi) \) is integrated over a control volume, and then Gauss theorem is used to convert it to a face integral. Its discretisation can also be obtained using equations (0.11) and (0.12) by replacing \( \sum \) with vector product:

\[
\nabla (\varphi) = \sum_{j \in S_i} S_f \varphi_f \tag{0.22}
\]

The gradient term \( \nabla (\varphi) \) described in this Section is an explicit term. It is mainly used to map a face-centred field to a source vector of one rank higher than the face-centred field.