University of Alberta

Mathematical Modeling, Experimental Testing and Numerical Simulation of Interfacial Coupling Phenomena of Two-Phase Flow in Porous Media

by

Oluropo Rufus Ayodele ©

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Petroleum Engineering

Department of Civil and Environmental Engineering

Edmonton, Alberta, Canada

Spring 2004

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 0-612-96236-9
Our file Notre référence
ISBN: 0-612-96236-9

ABSTRACT

Interfacial coupling in two-phase porous media flow was investigated analytically and numerically by comparison with experimental data. Modified forms of Darcy's equation, which incorporated interfacial coupling (viscous and capillary coupling) and hydrodynamic effects, were formulated. A numerical scheme was developed to solve the equations and was codified into a standalone numerical simulator using the Java^{TM} programming language. From the results of the analyses carried out, the parameter that controls the amount of viscous coupling was, theoretically and experimentally, found to have maximum values of 2 and 0.001, respectively, in order to account for the effect of viscous coupling.

A comparison of analytical and experimental results shows that the transport equations give a good description of flow. The viscous coupling and capillary coupling effects are very small and can be neglected in horizontal, cocurrent flow. In horizontal, countercurrent flow, the capillary coupling was found to have a more significant effect than viscous coupling, which can be neglected. The hydrodynamic effects are found to be insignificant in horizontal, cocurrent and countercurrent flow. For vertical flow, analytical results show that viscous coupling effects are insignificant. Due to the limitations and non-availability of a complete set of vertical flow experimental data, the applicability of the capillary coupling concept could not be verified fully using the modified set of transport equations. Hence, the modified set of transport equation is yet to be verified for vertical flow.

Results of numerical studies show that the modified set of transport equations gives a good description of horizontal, two-phase, porous media cocurrent flow. Numerical results also confirm that the viscous and capillary coupling effects and hydrodynamic effects are very small in horizontal, cocurrent flow. It can be concluded that if the experimentally determined countercurrent effective permeability values are used in simulating a cocurrent flow process, the countercurrent effective permeability values would have to be divided by the interfacial coupling

parameter, in order to obtain accurate predicted profiles. The reverse of the above inference is true if the experimentally determined cocurrent effective permeability values are used in the numerical simulation of a countercurrent flow.

DEDICATION

To the almighty God and my family, for being always supportive

ACKNOWLEDGEMENT

The work presented in this dissertation is made possible through the help, advice and words of encouragement of a lot of people. I thank Dr. Ramon G. Bentsen for his help, fatherly advice and academic guidance throughout the period on this research. I will ever remain grateful for what you taught me and the experiences and valuable skills I gained from you. I really appreciate all of what you have done for me. I thank Dr. Luciane B. Cunha for her help, words of encouragement and academic advice. I appreciate both of my supervisors efforts in reading through the manuscripts and making necessary corrections.

In addition, I thank other faculty members of the School of Mining and Petroleum Engineering of the University of Alberta. I particularly thank Dr. Samuel Frimpong, Dr. Ergun Kuru, Dr. Marcel Polikar, Dr. Jozef Szymanski, and the late Dr. Bodgan Lepski for all their words of encouragement and advice. I also thank all other members of the examining committee. I thank Mr. Vasily Prudky for his assistance in fixing the experimental equipment when the need arose.

I appreciate the friendly and moral support of my friends, Mr. Jian Gong, Mr. Bani Sharif and Mr. Dave Lawal. I also thank my wife, Chioma Ayodele, for her moral support and encouragement. The financial assistance of the sponsors of this research, Imperial Oil Resources Canada (a subsidiary of ExxonMobil) and the Natural Sciences and Engineering Research Council of Canada (NSERC), is acknowledged.

Lastly, but not the least, I thank the almighty God for seeing me through.

Oluropo Rufus Ayodele

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE[*]

ROMAN LETTERS

1	subscript for wetting (w) phase
2	subscript for nonwetting (nw) phase
a	dimensionless parameters that must be determined experimentally (Equation (4.10)) or (Equation (7.10))
A	cross-sectional area of porous medium
A_o	fitting coefficient (Equation (6.1))
A_c	area under capillary pressure curve = capillary pressure normalizing parameter
b	width of core
B_o	fitting coefficient (Equation (6.1))
$\bar{c}$	velocity of light
c	parameter that controls the amount of viscous coupling
c_1, c_2	c in the wetting and in the nonwetting phases, respectively
C	capacitance of the resonator
C^o	sum of pressure at the inlet and outlet ends at any point in time
$C(S)$	capillary term $\left(-\frac{1}{M_r}\lambda(S)F_1(S)\lambda^0_{r2}(S)\frac{d\pi_c}{dS}\right)$
$D(f_i)$	a function of three independent variables, f_{i-1}, f_i and f_{i+1}
f_i	fractional flow of phase i; i = 1,2
$f_{oi-1}, f_{oi}, f_{oi+1}$	initially assumed or previous iteration values of f_{i-1}, f_i and f_{i+1}, respectively
$\tilde{f}_{i-1}, \tilde{f}_i, \tilde{f}_{i+1}$	approximate solutions of f_{i-1}, f_i and f_{i+1}, respectively
f_f	fractional flow at the flood front
f_o	frequency response
fq	average frequency response

[*] *The equations and the calculated results in this document are presented in a consistent set of units; hence,*

$F_1(S)$	$\dfrac{M_r(S)\lambda_{r1}^0(S)}{M_r(S)\lambda_{r1}^0(S)+\lambda_{r2}^0(S)}$
g	acceleration due to gravity
h	thickness of core
I_{sr}	stability number $\left(\dfrac{V}{\lambda_{1r}}\left(\dfrac{M_r-1-N_g}{\sigma_e}\right)\left(\dfrac{{M_r}^{5/3}+1}{(M_r+1).({M_r}^{1/3}+1)^2}\right).\dfrac{4h^2b^2}{h^2+b^2}\right)$
$G(S)$	gravity term $\left(1-\dfrac{\bar{\lambda}(S)[1+(\rho_2/\Delta\rho'(S))(1-R_{12})]N_g\lambda_{r2}^0(S)}{R_{12}(S)M_r}\right)F_1(S)$
J	Jacobian matrix
Java^{TM}	trademark of Sun Microsystems Inc.
K_{abs} , K	absolute permeability of a medium
K_{eff} , K_i	effective permeability to phase i at a given saturation, i = 1,2
K_{ij}	generalized effective permeability for phase i; i,j = 1,2
K_{ref}	reference permeability = K_{2r} or K_{1r} or K_{abs}
K_{or}	relative permeability to oil at initial or irreducible water saturation
K_{wr}	relative permeability to water at residual oil saturation
K_{2r}	relative permeability to the non-wetting phase at initial or irreducible saturation of wetting phase
K_{1r}	relative permeability to the wetting phase water at residual saturation of nonwetting phase
K_{r2}	relative permeability to the nonwetting phase
K_{r1}	relative permeability to the wetting phase
l	length of the central conductor of the resonator
L, x	length of porous medium
LabVIEW^{TM}	trademark of National Instruments Corporation

the units of variables and parameters are not specified here.

LAGO	light atmospheric gas oil
LHS	left hand side of an equation
M_r	end point mobility ratio $\left(M_r(S) = k_{wr}\mu_o/\mu_w k_{or} = \lambda^\circ_{1r}/\lambda^\circ_{2r}\right)$
M	viscosity ratio $(M = \mu_o/\mu_w = \mu_2/\mu_1)$
N_c	macroscopic capillary number $\left(N_c = A_c k_{wr}/vL\mu_w = A_c\lambda^\circ_{1r}/vL\right)$
N_g	$\frac{\lambda^0_{1r}\Delta\rho' g \sin\Theta}{v}$
$N^{oi}(S)$	$\frac{1-\alpha_i(S)}{2}\frac{d\pi_c}{dS}$
P_i	pressure of phase i; i = 1,2
P_t	total pressure drop across the core
P_c	original Leverett's definition of capillary pressure = $P_2 - P_1$
P_d	pressure difference between two flowing fluids
P_d'	displacement pressure
P_i^0	pressure in phase i under conditions of static equilibrium, i = 1,2
P_i^1	small change in pressure in phase i due to flow of the phase, i = 1,2
q, Q	total flow rate
q_i, Q_i	flow rate of phase i; i = 1,2
R_{12}	function relating the pressure gradient in phase 1 to that in phase 2 = a weak function of normalized saturation = $R_{12} = 1 - a(1-S)$
$2/\bar{r}_m$	unit reciprocal length
$\bar{r}_m$	average macroscopic mean radius
RHS	right hand side of an equation
S	normalized saturation $= \frac{(S_1 - S_{1i})}{(1 - S_{2r} - S_{1i})}$
S_f	normalized saturation at the front
S_i	saturation of phase i; i = 1,2

S_{1r}, S_{1i}, S_{wi}	initial or irreducible saturation of the wetting phase
S_{2r}, S_{or}	residual saturation of the nonwetting phase
S^*	normalized inlet end saturation
$\hat{S}*$	calculated normalized inlet saturation, $\hat{S}*$, in the iterative scheme
S_1^*	specific saturation of the wetting phase
S_2^*	specific saturation of the nonwetting phase
SSCO	steady-state cocurrent
SSCT	steady-state countercurrent
t	time
USCO	unsteady-state cocurrent
USCT	unsteady-state countercurrent
v_1	total wetting phase flux (velocity)
v_2	total nonwetting phase flux (velocity)
v_{12}	amount of wetting phase flux that arises due to viscous coupling
v_{21}	amount of nonwetting phase flux that arises due to viscous coupling
v	total velocity
V_1	total wetting phase injection volume
V_2	total non-wetting phase injection volume
V	total injection volume
x	distance along the length of the core
Z_o	characteristic impedance

SUBSCRIPTS

c	capillary
bt	breakthrough
d	displacement

f	flood front
g	gravity
i	irreducible
n	normalized
w	wetting, phase 1
nw	nonwetting, phase 2
nwr	nonwetting residual
rnw	reduced (normalized) non-wetting
rw	reduced (normalized) wetting
wr	wetting at residual (irreducible)
or	oil at residual
r	residual

GREEK LETTERS

$\alpha = \alpha_i$	Interfacial coupling parameter = $\alpha_{vi} . \alpha_{ci}$, i = 1,2
α_{vi}	viscous coupling parameter, i = 1,2
α_{ci}	capillary coupling parameter, i = 1,2
μ_i	viscosity of phase i; i = 1,2
λ_i	effective mobility of phase i; i = 1,2 = $\frac{K_{ref} \cdot K_{ri}}{\mu_i}$
λ_{ij}	generalized mobility of phase i; i,j = 1,2 = $\frac{K_{ij}}{\mu_j}$
λ_c	$\lambda_{12} = \lambda_{21}$, cross mobility = generalized mobility (Equation (2.8))
λ_i°	effective mobility determined using cocurrent flow for phase i; i = 1,2
λ_i^*	effective mobility determined using countercurrent flow for phase i; i = 1,2
λ_{1r}°	effective mobility to wetting phase at residual saturation of non-wetting phase
λ_{2r}°	effective mobility to non-wetting phase at initial saturation of wetting phase

$\lambda_{r1}^{0}(S)$	$\dfrac{\lambda_1^0(S)}{\lambda_{1r}^0}$
$\lambda_{r2}^{0}(S)$	$\dfrac{\lambda_2^0(S)}{\lambda_{2r}^0}$
λ_m°	parameter to ensure dimensional consistency
ϕ	porosity of a porous medium
θ_i	angle between two directional vectors (i = 1,2, 3)
ε	dielectric constant of air
Θ	angle made by the direction of flow with the horizontal
τ	pore volume = dimensionless time = $\dfrac{qt}{A\phi L(1-S_{2r}-S_{1_i})} = \dfrac{vt}{\phi L(1-S_{2r}-S_{1_i})}$
$\breve{\tau}$	injected volume = $\int_0^{S*}\xi(S,\tau)dS = \int_0^{S*}\int_S^{S*}\dfrac{N_c C(S)}{f(S,\tau)-G(S)}\mathrm{d}S\mathrm{d}S$
ξ	normalized distance $= \dfrac{x}{L}$
ξ_f	normalized distance at the front of the two phase flowing region
ξ_b	normalized distance behind the two phase flowing region
ρ	density of fluid
$\Delta\rho$	difference in density with hydrodynamic effects $= \rho_1 - R_{12}\rho_2$
	$= (\rho_1 - \rho_2$, if $R_{12} = 1)$
$\Delta\rho'$	difference in density without hydrodynamic effects $= \rho_1 - \rho_2$
σ_e	$\dfrac{A_c\phi(1-S_{1i}-S_{2r})}{2/\bar{r}_m}$
ƛ(S)	average coupling coefficient $= (\alpha_1 + \alpha_2)/2$
$\pi_c(S)$	dimensionless capillary pressure $= \dfrac{Pc - P_d'}{A_c}$
ψ_i	potential of phase i; i = 1,2
ψ_d	potential difference between two flowing fluids

$\overline{\psi}_i$	average (or effective) potential of phase i; i = 1,2
$\pi_{\psi i}$	dimensionless potential of phase i, i =1,2 = $\frac{\psi_i}{A_c}$
$\nabla\psi_i$	$=\frac{\partial\psi_i}{\partial x}+\frac{\partial\psi_i}{\partial y}+\frac{\partial\psi_i}{\partial z}$, i = 1,2
ψ_t	total potential drop across the core
ψ_{1b}	potential drop across the core where only phase1 is flowing (region I), behind region II
ψ_{bf}	potential drop across the core where phases 1 and 2 flow (region II)
ψ_{2f}	potential drop across the core where only phase 2 is flowing (region III), ahead of region II beyond the front
ψ_t	total potential drop across the core
$\pi_{\psi t}$	dimensionless total potential drop across the core = $\frac{\psi_t}{A_c}$
π_{Pt}	dimensionless total pressure drop across the core = $\frac{P_t}{A_c}$
$\pi_{\psi 1b}$	dimensionless potential drop in phase 1 behind the front = $\frac{\psi_{1b}}{A_c}$
$\pi_{\psi 2f}$	dimensionless potential drop phase 2 ahead of the front = $\frac{\psi_{2f}}{A_c}$
$\pi_{\psi 12}$	dimensionless potential drop at the frontal region phases 1 and 2 flow = $\frac{\psi_{12}}{A_c}$
ϖ	error in the computation of τ, = $\frac{\hat{\tau}-\tau}{\Delta\tau}$.
ϑ	a constant for the iterative decrement or increment of $\hat{S}*$
$\frac{d\xi}{dt}$	$\frac{q}{A\phi L}\frac{\partial f_1}{\partial S}.\frac{1}{(1-S_{2r}-S_{i1})}$, equivalent representation of Equation (5.16)

$$\frac{d\pi_c(S)}{dS} \qquad \frac{1}{A_c}\left[\frac{\left[\left(1+cS+dS^2\right)\left(-a-2bS\right)\right]-\left[\left(a(1-S)+b(1-S^2)\right)\left(c+2dS\right)\right]}{\left(1+cS+dS^2\right)^2}\right] \text{..for Data Groups A and C}$$

where a, b, c and d are the fitting coefficients

CHAPTER 1

INTRODUCTION

1.1. Background

To maximize the recovery of hydrocarbon resources, appropriate surface gathering systems, surface facilities and drilling techniques must be put in place. One way to achieve this is by using the right tools to predict the recovery of hydrocarbons to guide in the optimal choice of such systems, facilities and techniques. The displacement of oil from porous media by water is a very important phenomenon in petroleum production in natural-water drives, water flooding and steam flooding in thermal recovery operations. Quite often, the predictions for such mechanisms and operations are based on the Muskat extension of Darcy's equation and the results from such predictions are used in the design of production systems, facilities and drilling techniques.

1.2. Statement of Problem

The conventional form of Darcy's equation, as modified by Muskat and Meres (1936) and Muskat et al. (1937) for multiphase flow does not give an accurate prediction of the recovery of hydrocarbons from petroleum reservoirs when applied to simulate flow in petroleum reservoirs. Researchers have demonstrated this in the past (Parker (1989), Muccino et al. (1998), Bentsen (1998a and 1998b), Ayub and Bentsen (1999), Rose (1999 and 2000) and some other papers cited in these references). Ayub and Bentsen (1999) provided a detailed literature review with regards to this. The Muskat extension of Darcy's equation has been used for years to develop commercial simulators used in predicting and history matching hydrocarbon recovery. It has been observed that predictions from such simulators have not always matched actual reservoir performance. Many researchers have pointed out that the main reason for this is that, in multiphase flow, the presence of one fluid affects the flow of the other fluids leading to interfacial coupling effects, which may have a considerable effect on fluid flow. This problem was probably first identified by Yuster (1951), when he pointed out the importance of the transfer of viscous forces across the fluid-fluid interfaces. This phenomenon is also called the Yuster effect.

Commercial simulators are based on an approach that ignores the effects of interfacial coupling. Recent attempts have been made to incorporate interfacial coupling effects especially in non-conventional recovery processes like steam-assisted gravity drainage - SAGD (Yuan et al. (2001)). Interfacial coupling effects refer to momentum transfer, also called viscous coupling, capillary coupling, and other unidentified coupling that comes into play between two or more fluids flowing simultaneously through porous media. At present, viscous and capillary coupling are accepted as constituting the major components of interfacial coupling. Viscous coupling refers to the viscous drag exerted by one fluid on the other fluid when they flow in the same porous medium and it is usually associated with the mobility of the fluids, while capillary coupling refers to the coupling that arises due to coupling of pressure across the interfaces of the fluids through the capillary pressure function (Bentsen (2001)).

The interfacial coupling effect is not only of importance in the recovery of conventional light crude oil but also in the recovery of heavy oil and bitumen from oil sand deposits. For example, Nasr et al. (2000) investigated the countercurrent aspect of the SAGD (steam-assisted gravity drainage) process. Part of the investigation involved the numerical simulation of the SAGD process and determination of the sensitivities of different parameters using the CMG STARS simulator, one of the widely used commercial simulators based on the extension of the conventional Darcy's equation to multiphase flow. They found out that there was a significant difference between the steam-water countercurrent and cocurrent relative permeability curves based on history matching of the experiments. According to the investigators, the magnitude of the countercurrent relative permeability, at a given saturation, was found to be always less than that of the associated cocurrent relative permeability, except at the initial gas saturation. When cocurrent relative permeability curves were used to predict the experimental results, the numerical model significantly over-predicted the experimental results.

Maini (1998) provided further support for this observation. He found that the measured relative permeability to heavy oil does not always give a good simulation result and he suggests it might be futile to determine the relative permeability of heavy oil. He also suggested several reasons why this might be so. One of these is the presence of the viscous coupling effect, which was not incorporated into the conventional multiphase flow equation. He noted that if the viscous coupling effect exists, it would cause lubrication of the non-wetting phase by a wet film, which under certain conditions can make the relative permeability of the non-wetting phase much higher than unity. He also pointed out that the lack of experimental data, and the complexities of the mathematical formulations, results in this effect being ignored in reservoir simulation studies. He suggested that the existence of such an effect would result in relative permeability curves that do not conform to the classical models. If the conventional method of using Darcy's equation as applied to multiphase flow to develop commercial simulators is correct, then the results from simulators should be very close to or accurately match experimental or field results, even with different flow morphologies. Also, if the conventional method is correct, the relative permeability curves of both countercurrent and cocurrent flow should be the same since the fluids are produced from a single reservoir with its own unique properties.

The relative permeability curves have a considerable effect on simulation studies. The simulation results are very sensitive to a small change in the relative permeability curves. When a history match is not satisfactory, the usual method is to continuously adjust the relative permeability until a suitable result is achieved. This shows clearly that the equations governing simultaneous flow of fluids through porous media are not well formulated. Transport models have been developed in the past (Hassanizadeh and Gray (1979a), Hassanizadeh and Gray (1979b), de la Cruz and Spanos (1983), Kalaydjian (1990), Bacri et al. (1990) and Rose (1990a)) to quantify these effects. According to Rose (1999), such transport models cannot generally be accepted unless verified by detailed experimental procedures and results.

1.3. Aims, Scope and Limitations of Research

Recently, at the University of Alberta, a modified set of transport equations (mathematical models) that incorporate the effect of interfacial coupling was developed (Ayub (2000), Bentsen (2001 and 2003c) and Ayub and Bentsen (2004)). Laboratory equipment was also developed to test the modified set of equations (Ayub (2000) and Ayub and Bentsen (2000 and 2001)). In developing the set of equations, a partition concept was introduced into Kalaydjian's transport equations while ignoring the effect of interfacial momentum transfer (viscous coupling). These authors argue, rightly or wrongly, that momentum transfer constitutes only a negligible part of the coupling that takes place in natural porous media. They buttressed their point of view with results presented by Zarcone and Lenormand (1994) and Rakotomalala et al. (1995) who argued that only a negligible amount of viscous coupling takes place in porous media.

Due to certain limitations in the design of the laboratory equipment used in the acquisition of data in their studies, not all of the experiments conducted to verify the set of equations were successful. As a result of this, only a partial verification of the existence of the interfacial coupling effect was achieved. Also, the non-inclusion of momentum transfer (viscous) coupling might be a good approximation, but incorporating this in the models would be better. If the viscous coupling is included, numerical simulation sensitivity analysis can be carried out to actually ascertain if the inclusion or non-inclusion has a significant effect on the relative permeability curves and hence help to determine if it constitutes a major or a negligible part of the coupling phenomena. Hence, the aim of this research study is to investigate further the effects of interfacial coupling and to develop a better understanding of the flow of immiscible fluids through petroleum reservoirs. Details of how this is to be achieved are given in problem definitions and objectives in Chapter 3.

1.4. Methodology

In achieving the above goal, a modified set of transport equations that incorporates interfacial coupling effects in porous media, two-phase flow was developed. The earlier developed equipment was slightly modified. Experimental testing data collected with the equipment and experimental data available in the literature were used to test the transport equation. A numerical simulator was developed to test the modified models as well as to carry out sensitivity analysis.

1.5. Contribution and Significance of Research

It is hoped that this research helps to resolve the issue of interfacial coupling phenomena in multiphase flow and eventually lead to better and more accurate ways of predicting the recovery of hydrocarbon resources from petroleum reservoirs. This would be of importance not only in conventional light hydrocarbon recovery (like recovery from a fractured-matrix formation or a solution gas drive reservoir) but also in the recovery of heavy oil and bitumen using the SAGD (Steam-Assisted Gravity Drainage) process where countercurrent flow plays a significant role.

1.6. Report Structure

Chapter 2 contains a review of the literature. Chapter 3 discusses the problems to be solved and the objectives of this research in detail. The theoretical formulation of the appropriate transport equations and interfacial coupling parameters are contained in Chapter 4, while the numerical scheme, solution procedure and programming techniques for the transport equations are given in Chapter 5. Chapter 6 contains a description of and modifications of the equipment used and the experimental procedure employed. Results are discussed and analyzed in Chapter 7, while conclusions and recommendations are provided in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

2.1. Conventional Darcy's Equation for Multiphase Flow

Muskat and Meres (1936) and Muskat et al. (1937) extended Darcy's equation for single-phase flow to the simultaneous flow of two or more fluids in porous media. In their approach, a more useful form of Darcy's law was obtained by assuming that a rock, which contains more than one fluid, has an effective permeability to each fluid phase and that the effective permeability to each fluid is a function of its saturation. The effective permeability of a rock to a fluid with which it is 100% saturated is equal to the absolute permeability of the rock. The effective permeability to each fluid phase is considered to be independent of the other fluid phases and the phases are considered to be immiscible.

If the relative permeability is defined as the ratio of effective permeability of a phase to a base permeability, Darcy's law may be restated for an unsteady flow system, which contains two phases, as follows:

$$v_1 = -\frac{kk_{r1}}{\mu_1}(\nabla P_1 - \rho_1 g \sin\Theta) \quad\text{.......................................(2.1)}$$

$$v_2 = -\frac{kk_{r2}}{\mu_2}(\nabla P_2 - \rho_2 g \sin\Theta) \quad\text{.......................................(2.2)}$$

In the equations above, the subscripts 1 and 2 represent the wetting phase and non-wetting phase, respectively. Note that k_{r1} and k_{r2} are the relative permeabilities to the two phases at the saturation of the phases within the rock. The full meaning of all the terms in the equations are given in the nomenclature. Also, three base permeabilities can be used. These are:

1) The absolute air permeability,

2) The absolute water permeability,

3) The effective permeability to the oil phase at the residual wetting phase saturation (Craig (1971)).

Darcy's law is the basis for almost all calculations of fluid flow within a hydrocarbon reservoir. In order to use the law, it is necessary to determine the relative permeability of the reservoir rock to each of the fluid phases; this determination must be made throughout the range of fluid saturations that will be encountered.

Since Leverett (1941) first published his classical paper on the capillary behaviour of porous solids, in which he developed the fractional flow equation for the first time, many investigators have studied the problems involved in measuring and predicting relative permeability. It was in this paper that Leverett proposed that porous media be represented by interfacial curved surfaces instead of the usual approach of representing porous media by a bundle of straight, cylindrical capillary tubes. Leverett did not deal with the concept of relative permeability in this paper. His work, and that of Muskat and Meres (1936), formed the basis of later efforts to develop the relative permeability concept, as applied to reservoir simulation studies (Chatenever and Calhoun (1952), Collins (1961), Lelièvre (1966), Bear (1972) and so on).

Relative permeability data are very important input in almost all multiphase flow studies. They are used as input in reservoir simulation for the prediction of productivity, injectivity and expected recovery from a reservoir. They are used in predicting various expected production scenarios in order to optimize well placement in the reservoir, and they can be used also to diagnose the formation damage expected under different conditions. A summary of the major results of research on the determination of relative permeability is presented in the following section.

2.2. Determination of Relative Permeability

In this section, various methods of estimating or determining rock (or reservoir) relative permeability as reported in the flow-through-porous-media literature are discussed. Generally, relative permeability measurement methods can be classified into three categories, namely: direct laboratory measurement techniques, mathematical model techniques, field data and simulation techniques.

Wycoff and Botset (1936) described the relative permeability concept for fluid flow through porous media by making use of an extended form of Darcy's law. A detailed review of the various methods was presented by Amyx et al. (1960), Honarpour et al. (1986), Manai (1991) and Ayub (2000). A brief description of the methods is given here with additional information from more recent investigators.

The methods presented here are for two-phase flow, which is the main focus of this research. Three-phase relative permeability experiments are extremely difficult to conduct, if not impossible. Three-phase relative permeabilities are often obtained by synthesizing data obtained from two-phase experiments. Hence, the methods presented here, even though they are for two-phase flow, can be used also to determine three phase relative permeability curves when they are employed to synthesize three-phase relative permeability curves.

The direct laboratory methods can be classified further into steady and unsteady state techniques, the centrifuge technique and the capillary pressure technique. In this section, all the methods (laboratory; mathematical; field data (production) and simulation techniques) are presented. It should be noted, however, that the direct laboratory method is the most commonly used and acceptable way of estimating relative permeability. The usage and acceptability of the other methods in day-to-day reservoir engineering and simulation studies is still the subject of current and on-going research.

2.2.1. Direct Laboratory Measurement Technique

2.2.1.1. Conventional Core Flooding Technique (Steady and Unsteady State)

In conventional core flooding experiments, the equipment used for the measurement of parameters (pressure, saturation, rate, flux, volume and weight) are designed in such a way as to minimize end effects at the ends of the core. This is a major problem in the laboratory measurement of relative permeability.

The **steady state method** is widely used because it is very reliable. Using this method, the wetting and the nonwetting phases are injected simultaneously at one end of the core and produced at the other end. The injection rates are maintained at a predetermined ratio so as to be able to achieve the desired saturation level. This is usually done through calibration of the pumping devices, or by trial and error. This method is time-consuming because a lot of time is required to achieve a uniform saturation level within the core. Once a uniform saturation level is achieved, the pressure, saturation, rate, flux, and weight (and/or volume) data are continuously monitored and recorded by the sensing devices.

The most commonly and widely used steady methods are: the Penn-state method, single-sample dynamic method, stationary fluid method, Hassler method, Hafford method and the dispersed feed method (Craig (1971) and Honarpour et al. (1986)). Several variations of these methods are still being developed in order to improve the overall accuracy of relative permeability measurements. Such variations and improvements include those presented by Manai (1991), Guo and Vatne (1993), Virnovsky, et al. (1995), Urkedal et al. (2000), Ayub (2000), Ayub and Bentsen (2000 and 2001), Dana and Skoczylas (2002) and so on. Such newer techniques also tend to make improvements in the experimental design procedures and sensing devices used in the measurement of the parameters, as well as the elimination of end effects. For example, the X-ray computer tomography (CT) scan technique is becoming more widely used instead of the microwave technique in saturation measurement. Also, more sensitive devices are becoming routinely used in pressure measurement. Many of these techniques have also been adapted for the measurement of heavy oil and bitumen (oilsand) relative permeabilities (Maini and Batycky (1985), Polikar et al. (1989), and Maini et al. (1990)).

Henderson et al. (2000) gave descriptions of the various techniques developed in the past for obtaining the relative permeability of gas condensate. These authors also developed a new reliable laboratory technique for obtaining gas-condensate relative permeability data under steady state conditions. This technique is an improvement over the previous techniques. They used condensing fluids as opposed to the general practice of using non-condensing fluids when obtaining the relative permeability of gas condensate. They used the steady state, gas-condensate and rate-sensitive relative permeability data to formulate a new correlation. The correlation relates gas and condensate relative permeability to the capillary number. A schematic representation of a typical laboratory two-phase relative permeability apparatus is shown in Figure 2.1.

The **unsteady state method** (also called the external drive method or the dynamic displacement method) is also widely used and it has the advantage that it is much faster than the time-consuming steady state method. With this method, the wetting phase is injected at a constant rate or variable rates into a nonwetting phase saturated core. Saturation equilibrium is not achieved. The volumes of injected and produced fluids are monitored and recorded along with pressure differences and saturation variations across the core. This method is considered to be more representative of reservoirs because saturation changes occur so quickly that equilibrium can hardly be attained (Polikar et al. (1989)). Although, this method is much faster than the steady state method, it has many operational problems such as viscous fingering, channeling and end effects. The interpretation of the method is also questionable in certain situations because of the heterogeneity of the porous medium, scaling requirements and flow instability (Peters (1979), Peters and Flock (1981), Bentsen (1985), Peters and Khataniar (1987), Sarma (1988)).

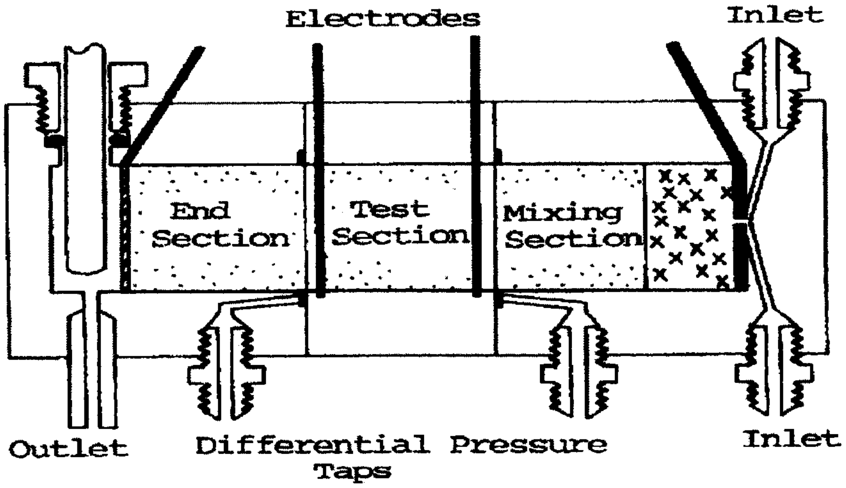


Figure 2.1: Schematic representation of a typical laboratory two-phase relative permeability apparatus with electrical resistivity saturation measurement (Courtesy, Honarpour et al. (1986) and Geffen et al. (1951)).

Almost all the unsteady state methods are based on the Buckley-Leverett (Buckley and Leverett (1942)) equation of frontal advance theory that neglects gravitational and capillary pressure effects. Several modifications of the Buckley-Leverett approach have been developed. The most common of these modified unsteady state methods are those developed by Welge (1952), Johnson et al. (1959), also called the BJN technique, and Jones and Roszelle (1978). Many more improvements have also been presented to improve the accuracy and reliability of the unsteady state methods. These improvements include the work by Naar et al. (1962), Sigmund and McCaffery (1979), Batycky et al. (1981), Huan and Shen (1982), Saraf and McCaffery (1982), Tao and Watson (1984), Islam and Bentsen (1986), Kerig and Watson (1986 and 1987), Civan and Donaldson (1987), Sarma and Bentsen (1989a, 1989b and 1990), Richmond and Watson (1990), Shen and Qing (1994) and King (1996). Modified unsteady state techniques for heavy oil and bitumen include those presented by Polikar et al. (1989), and Maini et al. (1990). Akin (2001) also proposed a new methodology for obtaining fracture relative permeability curves from unsteady state coreflooding experiments. Toth et al. (2002) developed convenient formulae for the determination of relative permeability from unsteady state two-phase fluid displacement under constant rate or pressure conditions. The formulae enable after-breakthrough processing of fluid displacement data. The formulae developed by Toth et al. allow direct and rapid determination of the relative permeability from unsteady state experiments.

2.2.1.2. Centrifuge Technique

Hassler and Brunner (1945) presented the theoretical evidence to support the use of the centrifuge technique to determine capillary pressure data. Hagoort (1980) was the first to report the experimental determination of relative permeability curves using the centrifuge method. This technique is relatively faster than the other laboratory techniques. Using this technique, there is no problem of viscous fingering, or instability, associated with the unsteady state experiments, but the capillary end effect is a big disadvantage.

With this technique, a small core plug is required. The core is held in a centrifuge without any confining overburden pressure. The wetting phase saturated core is placed in the core holder and centrifuged inside the centrifuge equipment at a constant speed with the nonwetting phase displacing the wetting phase. "The centrifugal force generated acts as an increased gravitational or buoyancy force and causes displacement" (Heaviside (1991)). Production from the core is monitored as a function of time. The fluids' relative permeability is determined from the production data. The mathematics of the centrifuge method was presented by Slobod et al. (1952), van Spronsen (1982), O'Meara and Leas (1983) and Kantzas et al. (1995).

In most situations, though not in all, the experiment is conducted as a gravity drainage experiment in which gas displaces oil (or water). As a result of this, the applicability of the technique to imbibition processes, which prevail in most reservoirs, has been questioned. Several other investigators have used this technique or made some improvements in the design of the equipment and experiment, analysis and interpretation of the centrifuge techniques. These improvements include work by Slobod et al. (1952), van Spronsen (1982), O'Meara and Leas (1983), O'Meara and Crump (1985), Munkvold and Torsaeter (1990), King et al. (1990), Firoozabadi and Aziz (1991), Hirasaki et al. (1992), Nordtvedt et al. (1993), Kantzas et al. (1995) and so on.

Honarpour et al. (1986), Heaviside (1991) and Kantzas et al. (1995) presented detailed reviews of the centrifuge techniques that have been employed in the determination of relative permeability. Ali and Barrufet (2001) also used the data obtained from an ultracentrifuge experiment to obtain the relative permeabilities before and after polymer flooding, making use of Wyllie and Gardner's model, to investigate the effects of polymer flooding on relative permeability curves. Figure 2.2 shows a schematic representation of a typical automated centrifuge apparatus.

2.2.1.3. Capillary Pressure Technique

This technique is only applicable to drainage processes. Hence, the applicability of this technique to imbibition processes, which prevail in most reservoirs, is also in doubt (Honarpour et al. (1986)). The capillary pressure technique is employed for a small core with a very low permeability on which the conventional flow or displacement test cannot be carried out. As reported by Honarpour et al. (1986), this technique is employed in situations where the conventional steady state and unsteady state techniques are not adequate. It is used, for example, in the estimation of the gas-oil relative permeability ratios in retrograde gas condensate reservoirs, where the saturation to oil increases as pressure decreases, with an initial saturation of oil as low as zero.

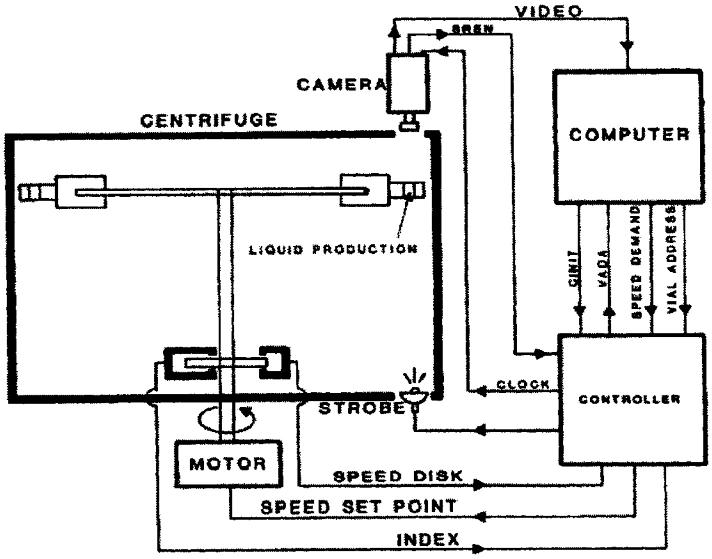


Figure 2.2: Schematic representation of a typical automated centrifuge apparatus
Courtesy: Honarpour et al. (1986)) and O'Mera and Leas (1983).

Capillary pressure and saturation data are collected as the non-wetting phase displaces the wetting phase. Mercury, air, brine or oil may be used as the nonwetting phase. Once the saturation and capillary pressure data are collected, equations developed by several investigators (Purcell (1949), Fatt and Dyksra (1951) and Burdine (1953)) are then used to determine the relative permeabilities. In these equations, the relative permeability is expressed in terms of the saturations, end point saturations and the change of capillary pressure with saturation.

2.2.2. Mathematical Model Technique

Several mathematical models have been developed to determine relative permeability for both two-phase and three phase flow. These mathematical models are based on correlations between the relative permeability and various data obtained from experimental measurements. Though they are not thought to be a good alternative to the laboratory measurements, they are used at times in order to avoid the complex processes used in laboratory studies. Also, they are complementary to the laboratory experimental data since they are often used to extrapolate limited laboratory data. The mathematical models are classified as follows.

2.2.2.1 Empirical Models

Empirical models are based on the empirical relationships between relative permeability and other measured parameters, most especially saturation. They provide the most useful approximations (Honarpour et al. (1986)). Examples of empirical models include Corey's model for drainage, Naar-Wygal's for imbibition, Naar-Henderson's model for imbibition and several other models reported by Honarpour et al. (1986) and Honarpour and Mahmood (1988). In addition to two-phase empirical models, three phase empirical models have also been developed. Stone (1970 and 1973) developed generally accepted empirical models for three-phase flow and subsequent research efforts in this area have focused on modifications of Stone's methods. A summary of these modifications is presented by Blunt (2000).

Recently, a new empirical model was developed for three-phase relative permeability by Blunt (2000). The method incorporates changes in the composition of the hydrocarbon. The method tried to eliminate the shortcomings of previous approaches such as those of Stone and others. The model was based on saturation-weighted interpolation between the two-phase relative permeabilities that was first proposed by Baker in 1988. The model can predict the behaviour for any sequence of saturation changes and account for oil and gas trapping as well for oil layer drainage. For any given set of two-phase measurements, the model can predict the three-phase relative permeability for any saturation, wettability and hydrocarbon composition.

2.2.2.2 Capillary Models

Capillary models use the assumption that the porous medium is made up of bundles of capillary tubes of different diameters. Capillary models do not provide realistic results because they assume the absence of channels in porous media; that is, they ignore the interconnected nature of the porous medium. Purcell (1949) developed a capillary model for estimating the relative permeability of porous media. The model uses capillary pressure and porosity to determine relative permeability.

2.2.2.3 Statistical Models

Statistical models are also based on the assumption of capillary tubes. The models assume that the porous medium is made up of bundles of capillary tubes with diameters that are randomly distributed; hence, they do not provide too realistic results.

2.2.2.4 Network Models

Network models are used to describe the dynamics of fluids flowing in porous media at the microscopic level. Network models are becoming commonly used in the determinations of relative permeability because they provide a quantitative description of porous media at the micro pore level. Network models are based on the analogy of electric resistor systems. They provide the best way to model the dynamic behavior of porous media (Hassler et al. (1936) and Honarpour et al. (1986)).

Heiba et al. (1984), Bryant and Blunt (1992), Fenwick and Blunt (1998a and 1998b), Mani and Mohanty (1998), Øren et al. (1998), Lerdahl et al. (2000) and Bèkri et al. (2001) have all developed different or similar pore network models that predict relative permeability. The technique developed by Bèkri et al. is applicable to a system in which there is a phase change of liquid to gas and retrograde condensate. These network models are based on an understanding of the pore structure and physics of flow. Lerdahl et al. (2000) and Maximenko and Kadet (2000) presented detailed reviews of the relative permeability networks models that have been developed in the past.

Lerdahl et al. (2000) and Maximenko and Kadet (2000) also developed new network models for obtaining the relative permeability for multiphase systems. Lerdahl et al. (2000) developed a predictive network model for describing the intrinsic properties of a reservoir rock. The model simulates two-phase flow and drainage dominated flows. Relative permeability computed with the network model for the reconstructed sample rock (sandstone) compares favourably with the experimental results for both two-phase and three-phase systems. Maximenko and Kadet (2000) developed a network model for obtaining relative permeability that does not include the interfacial coupling effect. They model the porous medium as a network of capillaries. They used the percolation and effective-medium models to calculate the network conductivity. They also used the model to determine the efficiency of a polymer flooding system.

2.2.3. Field Data and Simulation Techniques

2.2.3.1 Field Data Technique

Production, pressure and saturation history or data from a field can also be used to determine the relative permeability. This process is an inverse approach. The rate, pressure and saturation data from the field are substituted into Darcy's equation (modified for multiphase flow, that is, the Muskat extension). This approach is not very accurate because it gives poor agreement with relative permeabilities obtained in the laboratory. Possible sources of error include the mechanical skin effect, assumption of constant gas-oil ratio, encroachment of water into the reservoir from casing leaks and fractures which interfere with the oil-water ratio, use of data from a well that may not give actual reservoir responses and so on.

Apart from the direct use of the conventional Darcy's equation, other techniques such as artificial intelligence techniques (neural network and genetic techniques) or streamline-based approaches may be used also to back out relative permeability from field production, pressure and saturation data.

Using neural network or generic algorithm techniques (artificial Intelligence techniques) is also an inverse based system. The neural network (singly), or combined with a genetic algorithm technique (hybrid system), is trained using the field data from some part of the wells. The trained network is then used to predict the relative permeability for the whole reservoir or field. A neuro-simulation technique was recently developed by Al-Wadahi et al. (2000). Akin and Demiral (1998) also developed a genetic algorithm technique for estimating permeabilities from a displacement experiment.

The streamline approach is another inverse based approach. Kulkarni and Datta-Gupta (2000) presented a field method, called a streamline-based approach, of estimating the relative permeability from production data. The method allows analytical computation of the sensitivity of the production response with respect to relative permeability. It requires a single streamline simulation run and an analogy can be drawn between streamline and seismic ray tracing to develop a good mechanism for inversion of production data. Power function and B-spline representations of relative permeability were developed. Pressure data and water cut data were used to determine simultaneously the absolute and relatively permeability.

2.2.3.2 Simulation Technique

Numerical Simulation techniques are basically a history matching process, which is similar to the calculations from field data. But rather than using an analytical equation, a numerical simulator or numerical solution technique is employed to calculate the relative permeability from the field data or laboratory data. Many investigators have presented work that uses the numerical solution methods to solve the two-phase flow equations in core flooding experiments (Archer and Wong (1973), Batycky et al. (1981), Fassihi (1989), and Maini et al. (1990)). Akin (2001) also used matrix relative permeability data to obtain the fracture relative permeabilities by history matching experimental data with double porosity numerical models.

Pressure and saturation data (either from the field or laboratory) are fed into the simulator and they are used to obtain history-matched relative permeabilities. Details of such an approach are well described by Heaviside et al. (1988) and Allen et al. (1982). "Unique results are not always possible, and, depending on the nature of the experiment, the history match may not be sensitive to some portions of the input data" (Heaviside (1991)). This is a major limitation of this approach. The application of numerical techniques gives a more complete analysis of laboratory displacement experiments, but it often requires cumbersome capillary pressure measurements.

2.3. Factors Affecting Relative Permeability.

Several factors or variables affect relative permeability. These include: saturation, saturation history, preferential wettability of the rock to one of the fluids, rock properties, flow rates and regimes, viscosity and viscosity ratio, interfacial tension and density, over burden pressure, displacement pressure, pressure gradient, initial wetting phase saturation, porosity and permeability, temperature, and so on. Detailed reviews of these various factors or variables are presented by Scheidegger (1974), Honarpour et al. (1986), Islam and Bentsen (1987), Manai (1991), Ayub (2000) and several other investigators.

Despite such dependencies, laboratory experiments are conducted in such a way that they are made to be dependent solely on saturation. Although it is usual to assume in reservoir engineering that relative permeability is solely dependent on saturation, this assumption is actually an approximation (Scheidegger (1960)) and this is generally accepted. "Only by restricting the applicability of a given relative permeability curve to a kind of rock, a class of fluids, a type of flow regime, a range of flow rates and an interval of temperature can it be said that the relative permeability is a function of saturation only. However, saturation is a decisive variable" (Morel-Seytoux (1969)). Also, a quantitative relationship between these variables and relative permeability is not well established and it is usual to state the experimental conditions such as steady state, unsteady state, drainage and imbibition when measuring the relative permeability as a function of saturation alone.

2.4. Interfacial Coupling Fundamentals in Multiphase Flow Transport Equations

2.4.1. Background

The concept of two or more fluids flowing simultaneously in porous media has been treated extensively in a number of monographs (Collins (1961), Childs (1969), de Wiest (1969), Craig (1971), Bear (1972), Scheidegger (1974) and Sahimi (1995)). This concept has also been treated in several papers (Muskat and Meres (1936), Muskat et al. (1937), Morel-Seytoux (1969), Philip (1970 and 1973), Wooding and Morel-Seytoux (1976), Kalaydjian (1987), Greenkorn (1983), de la Cruz and Spanos (1983), Whitaker (1986), Kalaydjian and Legait (1987), Bentsen and Manai (1991 and 1993) Bentsen (1998a and 1998b), Ayub (2000), and so on). According to Bear (1972), the flow of immiscible fluids in porous media may be conveniently subdivided into two categories: (1) steady state (that is, all macroscopic properties of the system are time invariant at all points); and (2) unsteady state (that is, properties change with time).

In the steady state flow of immiscible fluids, the saturation of the medium with respect to all the fluids contained in the system is constant at all (macroscopic) points. Hence, in steady flow there is no displacement of any fluid by any of the other fluids in the pores. On the other hand, in unsteady state flow, the saturation at a given point in the system will, in general, change. Therefore, displacement phenomena fall in this category.

One can distinguish also cocurrent flow, when both phases flow in the same direction and countercurrent flow, when different phases flow in opposite directions. The latter takes place, for example, when a system saturated with a nonwetting phase is brought in contact with a wetting fluid. The latter will imbibe and drive out some of the nonwetting fluid in countercurrent flow (Graham and Richardson (1959)). Another example is when capillary forces are small as compared with gravitational and viscous forces, which is the case when heavy oil drains from a formation in countercurrent flow with steam.

Over the years, various investigators have come to realize that the extension of the conventional Darcy's equation to multiphase flow might not actually be a correct way of mathematically describing multiphase flow. The usual Muskat's equation was developed for cocurrent flow and not for countercurrent flow. In this regard, several schools of thought have evolved.

Muskat's extension of Darcy's equation is widely known. This equation is widely used in reservoir simulation including condensate reservoirs (for example, a work reported by Khuzhayorov and Burnashev (2001) on oil-gas-condensate modeling). In this school of thought, the flux in a given phase is taken to be proportional to the driving force or potential gradient acting across the phase. The mathematical expressions of this statement are given in Equations 2.1 and 2.2. Several studies have been carried out using this equation and almost all commercial numerical simulator codes are based on this general simplification.

Even in situations where countercurrent flow predominates, such as in highly fractured reservoirs, this equation is widely used with adjustment of the boundary conditions (Blair (1960)) and the use of derived analytical models (Chen et al. (1995) and Cil et al. (1998)) to account for the counter current flow situations. Such analytical models (also called matrix/fracture transfer functions) are used to describe flow in which countercurrent flow dominates because the physics of counter current flow is not well understood. Cil et al. (1998) provided details of these functions, which they classified into four main categories: diffusivity models (equations), scaling relations, empirical models and material balance models. This Muskat approach seems not to be a very good way to model the correct physics of multiphase flow and several investigators later show this approach to be inaccurate. Notable researchers (de la Cruz and Spanos (1983), Bentsen (1998a and 1998b), Rose (1988, 1999 and 2000) and so on) have shown that the Muskat approach is an over simplification of the problem at hand.

Yuster (1951) was the first to identify what has been termed the interfacial coupling effect in multiphase flow. Yuster identified this effect as the transfer of viscous forces across the fluid-fluid interfaces. This phenomenon is also called the Yuster effect. Scott and Rose (1953) later investigated this phenomenon and they term this effect "viscous coupling". Klute (1967), Bolt and Groenevelt (1969) and Bear (1972) presented further evidence to prove the existence of viscous coupling.

2.4.2. Quantification and Understanding of Interfacial Coupling Effects

A summary of the theoretical, experimental and numerical attempts to modify Muskat and Meres (1936) and Muskat et al. (1937) extension of Darcy's equation to account for interfacial coupling effects is available (see Ayub and Bentsen (1999), Rose (1988, 1999 and 2000)). Research in this area has been conducted over the past 4 decades. A brief description of these research efforts together with more recent efforts is given as follows.

2.4.2.1. Theoretical Approach: Volume Averaging and Irreversible Thermodynamics

Several theoretical approaches have been used to modify Muskat's extension. The most common and popular of these approaches include the volume averaging techniques and the use of Onsager's reciprocity principle, which is based on irreversible thermodynamics (Onsager (1931a and 1931b)).

Volume averaging technique approaches include work by (Whitaker (1967, 1973 and 1986), Slattery (1969 and 1970), Bear (1972), Gray (1975), Trapp (1976), Gray and Lee (1977), Hassanizadeh and Gray (1979a, 1979b and 1980), Narasimhan (1980), de la Cruz and Spanos (1983), Bachmat and Bear (1986), and Bear and Bachmat (1986)).

Studies done using the principles of irreversible thermodynamics to arrive at a modified form of these equations, which incorporate the viscous or coupling terms, include those by (de Groot (1963), de Groot and Mazur (1963), Katchalsky and Curran (1967), Bear (1972), Kalaydjian (1987 and 1990), Kalaydjian and Marle (1987), Longeron (1987), and del Riop and de Harro (1992)). They used the basis of symmetry and results arising out of Onsager's irreversible thermodynamic principle to modify Muskat's equation.

The modifications based on volume averaging and Onsager's reciprocity relations include a cross, or coupling, term that is proportional to the potential gradient or pressure gradient of a given phase. Also, the flux in a given phase is proportional to the potential gradient in the phase and the potential gradient in the other phase. This approach has been the subject of intense research over a period of more than four decades and landmark contributions in this school of thought are those cited above. These modified equations, with the generalized phase permeabilities, are given as:

$$v_1 = -\frac{kk_{r11}}{\mu_1}\frac{\partial\psi_1}{\partial x} + \frac{kk_{r12}}{\mu_2}\frac{\partial\psi_2}{\partial x} \qquad (2.3)$$

$$v_2 = -\frac{kk_{r21}}{\mu_1}\frac{\partial\psi_1}{\partial x} + \frac{kk_{r22}}{\mu_2}\frac{\partial\psi_2}{\partial x} \qquad (2.4)$$

The meaning of the terms in Equations 2.3 and 2.4 are given in the nomenclature.

2.4.2.2. Theory of Mixtures and Network Models

Some other techniques have used the theory of mixtures and network models. Work based on the theory of mixtures includes work presented by Drumheller (1978), Bowen (1980 and 1982), Bedford and Drumheller (1983), Murdoch and Kowalski (1992), Wang and Beckermann (1993), Wang and Cheng (1996), Cheng and Wang (1996), Wang (1997) and Rose (1990a and 1993). Goode and Ramakrishnan (1993) used the theory of network models to arrive at a new set of transport equations.

Goode and Ramakrishnan (1993) proposed a new set of multiphase transport equations using a network model based on a four-cusp pore. A simple cubic network model of 30x30x30 bonds was used. The network model was used to simulate the invasion of the nonwetting phase using a percolation-like algorithm. The form of the new set of equations proposed by Goode and Ramakrishnan is given as:

$$v_1 = -\frac{kk_{r1}(S_1,M)}{\mu_1}\frac{\partial P_1}{\partial x} - \frac{kk_{rc}(S_1,M)}{(\mu_1+\mu_2)}\frac{\partial P_2}{\partial x} \qquad (2.5)$$

$$v_2 = -\frac{kk_{rc}(S_1,M)}{(\mu_1+\mu_2)}\frac{\partial P_1}{\partial x} - \frac{kk_{r2}(S_1,M)}{\mu_2}\frac{\partial P_2}{\partial x} \qquad (2.6)$$

where

$$k_{rc} = \frac{\lambda_c(\mu_1+\mu_2)}{k} \qquad (2.7)$$

The term M is the viscosity ratio and the term, λ_c, is the cross term mobility and is defined as:

$$\lambda_c = \lambda_{12} = \lambda_{21} \qquad (2.8)$$

From Equations 2.5, 2.6, 2.7 and 2.8, and the flow calculation analysis performed by Goode and Ramakrishnan, it can be concluded that the nonwetting phase relative permeability has a strong dependency on saturation and viscosity ratio. Also, the off-diagonal term was shown to contribute to a fraction of the flow. They, however, found that relative permeability of the wetting phase shows a weak dependence on viscosity ratio, which is in line with conventional wisdom.

2.4.2.3. Numerical Simulation Techniques

Conventional simulation techniques, and lattice gas automata simulation techniques, have been used also to investigate and verify the viscous coupling effect. Work done using conventional simulation techniques includes that by Rose (1990b), Goode (1991), and Goode and Ramakrishnan (1993). The lattice gas automata technique, a computational fluid dynamics (CFD) technique, has been extended to describe single and multiphase fluid flow in general and, specifically, fluid flow in porous media.

Notable work done using the Lattice gas automata technique or its modifications include work presented by Hardy et al. (1976), Frisch et al. (1986 and 1987), Rothman and Keller (1988), Kadanoff et al. (1989), Rothman (1988 and 1990), Gutman (1990) and Chen et al. (1992). Others studies include, the work by Gao and Sharma (1994a and 1994b), Ferreol and Rothman (1995), van Genabeek and Rothman (1996), Oslon and Rothman (1997), Satofuka and Nishioka (1999), Küntz et al. (2001) and Santos et al. (2002).

Kalaydjian and Marle (1989) also presented a general overview and critique of several techniques of modeling multiphase flow in porous media. These techniques included, among others, theoretical continuum mechanics descriptions at the pore level, capillary tube modeling, network modeling, statistical techniques, volume averaging and the multi-scale methods (similar to volume averaging) discussed above, as well as smaller (molecular) level methods - lattice gas automata and molecular dynamics approaches. They referred to these molecular level methods as discrete modeling methods of multiphase flows. They stated that these discrete methods offer new approaches to the discussion of the boundary conditions that are normally used in the macroscopic description of multiphase flow and may provide further explanation of the complicated nature of multiphase flow in porous media.

The lattice gas automata (LGA) method, which has been modified as the Lattice-Boltzmann method and later as the Lattice BGK method (Bhatnagar et al. (1954), Qian (1990) and Qian et al. (1992)), is based on the lattice-gas formalism. It can be used to describe 2-dimensional and 3-dimensional multiphase flow problems. In this method, "*the discrete structure of the fluid and solid phases are taken into account and their velocities are such that at each time step, they move from the initial position to one of the adjacent vertices. Collisions occur only at the vertices. Knowing the velocities of the colliding particles before a collision, given rules are applied to determine the velocities after the collision, hence, the vertices where the particles will go at the next time. These rules must satisfy the conservation of total momentum and energy, and are chosen in such a way that the macroscopic behaviour of the model is similar to that of the real fluid*" (Kalaydjian and Marle (1989)).

The best way towards making results of lattice gas automata and its various modifications more reliable is to use a detailed image of a porous medium captured through a nuclear magnetic resonance (NMR) tool or other sources as input into the lattice-gas-based simulator. Such an image gives accurate information about the porous medium matrix, hence providing a basis for accurate prediction. Manz and Gladden (1999) used such an approach. They used the Lattice-Boltzmann method to simulate the single-phase flow and dispersion in porous media and they compare the results to that of NMR experimental studies. Manz and Gladden used a magnetic resonance image visualization of three-dimensional packing of spheres as the matrix for the simulation. They obtained quantitative agreement between the NMR measurements and the predictions of the Lattice-Boltzmann simulation. However the approach employed by Manz and Gladden is quite expensive in terms of the availability of NMR resources to capture the whole reservoir volume. Hopefully, seismic images, although with much smaller resolution, might provide alternative options. The most widely used option is to generate randomly lattice points through a statistical approach, that is, assumption of uniform lattice nodes.

As part of a two-dimensional Lattice-Boltzmann simulation study of a three-phase immiscible flow at the pore scale, van Kats and Egberts (1999) studied the viscous coupling effect in porous media. They simulated steady-state three-phase flow, at constant and equal saturation, in a small segment of a water-wet porous medium under spreading and non-spreading conditions. They found out that the coupling coefficients are not significantly affected by the value of the spreading coefficient (a measure of the differences between interfacial tension for each pair of the three phases). For a spreading condition at steady state, the value of the spreading coefficient is zero and for a non-spreading condition at steady state, the value of the spreading coefficient is negative. They also confirmed the applicability of the Onsager-Casimir reciprocal relation in the non-spreading case and to a lesser extent in the spreading case. They recommended further investigation on the generality of these results.

Similarly, Langaas and Papatzacos (2001) numerically investigated the steady state relative permeability of a simplified porous medium using the Lattice-Boltzmann method. They investigated the dependence of cocurrent and countercurrent flow relative permeabilities on phase saturation, wettability, driving forces and viscosity ratio. They found out that countercurrent relative permeabilities are always less than the cocurrent ones, as generally observed in laboratory experiments and they attributed this to the opposing effect of the viscous coupling, and partly to the different levels of capillary forces.

Malevanets and Yeomans (2000) defined a lattice Boltzmann algorithm for two-phase flow that incorporates viscous coupling between the two fluid phases. From the results of numerical testing, they found that when oscillatory shear is applied to one phase, a vortices' pattern is set up which leads to concentration (ratio of a phase density to total density) modulations. They concluded that the "concentration modulation is driven by a steady flow structure imposed on the fast oscillatory flow". The modulations can either be regular or chaotic depending on the fluid free energy form and the relationship between viscosity and concentration - a high dependence of viscosity on concentration favours high modulations. They suggested that the pattern formation might be due to the reduction of viscous drag by the fluid, leading to entropy production by undergoing phase segregation.

2.4.2.4. Experimental Techniques

Laboratory experimental attempts have focused on the quantification and verification of the viscous effect (Bourbiaux and Kalaydjian (1990), Bentsen (1998a and 1998b), Zarcone and Lenormand (1994), and Zhou et al (2002)). Most recent experimental (and theoretical and numerical) research efforts that attempt to quantify the coupling effect show the effect to be insignificant in cocurrent flow (Zarcone and Lenormand (1994), Rakotomalala et al. (1995), Wang (1997), and Dana and Skoczylas (2002)). Zarcone and Lenormand used an experimental approach, Rakotomalala et al. used analytical and lattice gas simulation approaches and Wang used the theory of mixtures to arrive at the same conclusion. Dana and Skoczylas also used experimental methods. Based on their experimental method, they concluded that viscous coupling effects are negligible. Results from this school of thought show that while interfacial coupling does exist, it is small and can be neglected as is done under the Muskat formulation.

2.4.2.5. Cocurrent and Countercurrent Dilemma

A much bigger problem arises when cocurrent and countercurrent experiments were conducted under the same conditions. The observed relative permeability curves in cocurrent flows had higher magnitudes than those for countercurrent flow (Lelièvre (1966), Bourbiaux and Kalaydjian (1990), and Bentsen and Manai (1991 and 1993)). Oil recoveries in cocurrent flow have also been reported to be higher than that in counter current flow (Naser et al. (2000) and Zhou et al. (2002)). One explanation for the difference in magnitude of cocurrent and countercurrent relative permeabilities is that the equations that relate the relative permeabilities (and hence the mobilities) to other measured properties do not capture the correct physics of flow. This is because such equations do not include appropriately the effect of interfacial coupling between the two phases. Interfacial coupling has been identified in both cocurrent and countercurrent flow. It can be concluded from the experimental work of Lelièvre (1966), Bourbiaux and Kalaydjian (1990) and Bentsen and Manai (1991, 1993) that the interfacial coupling effects are significantly larger in a countercurrent flow experiment. When interfacial coupling effects arise in porous media, proper association of this effect with either the pressure/potential gradient or the mobility is required. Hence there is a need to resort to a more rigorous approach.

From the explanation above, it can be concluded that another type of coupling, which might be of great importance in countercurrent flow, does exist. This new type of coupling is called capillary coupling (Babchin and Yuan (1997), Bentsen (2001), Ayub (2000) and Bentsen (2003b)). Langaas and Papatzacos (2001) also concluded from Lattice-Boltzmann numerical simulation studies that the relative permeabilities in countercurrent flow are always less those of cocurrent flow due partly to different levels of capillary forces. Capillary coupling arises due to the coupling of pressure across the curved interfaces in porous media. Hence, two types of coupling can be identified, namely, the conventional widely accepted coupling (viscous coupling or momentum transfer) and the newly proposed type of coupling (capillary coupling).

2.5. Viscous and Capillary Coupling Equations

Previous attempts to modify the multiphase equations include only the possibility of viscous coupling taking place. New formulations should be able to include the possibility of viscous coupling and capillary coupling taking place simultaneously.

Using channel flow theory and Onsager's reciprocity relation, Ayub (2000) and Bentsen (2001) proposed modified types of transport equations that incorporate capillary coupling for one dimensional flow. These equations are given as:

$$v_1 = -\lambda_1^0\left(\frac{\partial P_1}{\partial x} + \frac{1-\alpha_1}{2}\frac{\partial P_c}{\partial x}\right) \tag{2.9}$$

and

$$v_2 = -\lambda_2^0\left(\frac{\partial P_2}{\partial x} - \frac{1-\alpha_2}{2}\frac{\partial P_c}{\partial x}\right) \tag{2.10}$$

where $\lambda_i^* = \alpha_i \lambda_i^0$, and where α_i is the interfacial coupling parameter; which is defined as being equal to $(1-\phi)$, the capillary coupling term (Ayub (2000)). The meaning of all the terms in Equations 2.9 and 2.10 are given in the nomenclature. These equations do not incorporate viscous coupling since it was assumed to be negligible based on the previous work of Zarcone and Lenormand (1994) and Rakotomalala et al. (1995).

The equations partially solve the problem of identifying and quantifying interfacial coupling in multiphase flow through porous media by incorporating the effect of capillary coupling and showing that it is the capillarity of the porous medium that plays a more important role than viscous coupling. Although this is a modified and improved approach, it has the following disadvantages:

1) Viscous coupling was neglected in the formulation because experimental evidence supports this (Zarcone and Lenormand (1994)).
2) Hydrodynamic effects are neglected, that is, $R_{12} = 1$. (see defining equation for R_{12} in the nomenclature).
3) The formulation was done in terms of pressure and not potential.
4) Definition of capillary coupling did not account for the forced flow condition of the experiment.
5) Also, the equation was not verified for countercurrent flow situations.

Thus, there is a need to improve these equations in order to incorporate all of these effects, so that sensitivity analysis can be carried out to see the effect of neglecting any of the identified coupling effects in the experimental testing or verification stage.

2.6. Interfacial Coupling and Transport Equations for Heavy oil and Bitumen Recovery

The previous sections deal exclusively with the work carried out with conventional light hydrocarbons in which oil is produced through vertical, directional or horizontal wells. In this section, work reported in the literature, which specifically deals with steam-assisted gravity drainage (SAGD), a thermal recovery process is presented.

The SAGD method is increasingly becoming the preferred method of bitumen and heavy oil recovery. The importance of treating the SAGD transport equation separately is quite obvious. The SAGD process, in many respects, is different from the recovery methods used in the recovery of conventional light oil that is buried further down in the earth's crust. Bitumen and heavy oil are located at shallower depths and steam is required to initialize their movement from within the formation. Details of the SAGD process are well documented by Butler (1994).

Babchin et al. (1998) developed transport equations for the SAGD process. As noted by Babchin et al., in gravity driven processes like SAGD, the flow is more complex, consisting of cocurrent and countercurrent flows. Their work describes the characteristics of gravity driven flow and provides generalized permeabilities (or mobilities) for such flow. They mentioned that recent theoretical work has shown that both viscous coupling and capillary coupling play a significant role in multiphase flow and that both couplings give rise to non-zero cross coefficients in the transport equations. Despite this, their paper only discussed the application of a matrix formulation of the transport equations to the description of gravity drainage processes. No experimental validation was undertaken and the components of the equation that are responsible for viscous coupling and capillary coupling were not identified or partitioned. Details of their work are presented below.

In the SAGD recovery process, a layer of condensate a few centimeters thick separates the steam chamber and the unheated bitumen. Hence, at the edge of the steam chamber, the transport system is reduced to the flow of two incompressible phases. The temperature at the edge is lower than the rest of the steam chamber. Thus, the viscosity of bitumen on the edge is higher than that within the chamber. The rate of oil recovery may therefore be controlled by the combined bitumen-condensate flow at the edge. The two phases, bitumen and condensate (water) have cocurrent and countercurrent flow components. The cocurrent components are primarily controlled by gravity while the counter-current components are due to the combined action of the capillary pressure gradient and buoyancy forces (Babchin et al. (1998)). Fluxes of both fluids were expressed through relative permeability theory. Babchin et al. observed that the current formulation of numerical simulators does not operate by capillary pressure gradient and buoyancy forces, but rather by phase pressure gradients and gravity forces. Hence, they proposed equations in which the SAGD process fluid's fluxes would have to be expressed in the matrix formulation. The equations proposed by Babchin et al. are stated as follows:

$$v_1 = -A_{11}\frac{\partial\psi_1}{\partial x} - A_{12}\frac{\partial\psi_2}{\partial x} \qquad (2.11)$$

and

$$v_2 = -A_{21}\frac{\partial\psi_1}{\partial x} - A_{22}\frac{\partial\psi_2}{\partial x} \qquad (2.12)$$

where

$$A_{11} = \frac{1}{2}\left(\lambda_1^0 + \lambda_1^*\right) \qquad (2.13)$$

$$A_{12} = \frac{1}{2}\left(\lambda_1^0 - \lambda_1^*\right) \qquad (2.14)$$

$$A_{21} = \frac{1}{2}\left(\lambda_2^0 - \lambda_2^*\right) \qquad (2.15)$$

$$A_{22} = \frac{1}{2}\left(\lambda_2^0 + \lambda_2^*\right) \qquad (2.16)$$

From the equations above, it can be seen that the formulation simultaneously accounts for cocurrent and countercurrent components in the complex three-dimensional flow. Although this approach might seem plausible, it does not enable the quantification of the fluxes in cocurrent and countercurrent flow separately. The coupling effects in both cases (cocurrent and countercurrent flows) are lumped together into a set of coefficients represented by Equations 2.13 and 2.16, which are assumed to represent the real reservoir relative permeabilities.

Expanding on the work presented by Babchin et. al (1998), Yuan et al. (2001) developed a more generalized description of relative permeability matrix coefficients for SAGD based on experimental data reported in the literature. This generalization allows a choice of various independent pairs of relative permeability curves as input for a reservoir simulator (CMG's STAR simulator). They also generalized this concept to three-phase flow, which was implemented in CMG's STAR simulator.

2.7. Concluding Remarks

Several research efforts are still going on as to how interfacial coupling can be incorporated into the measurement of relative permeability values used in predicting recovery from conventional oil recovery and thermal recovery processes, especially the SAGD process. Most of the work done with light oil can be adapted to heavy oil and bitumen, if gravity, buoyancy and temperature effects are properly accounted for in the formulation process. It is hoped that the modified transport equations developed in this research work can be adapted for use in the measurement of the relative permeabilities used in predicting recovery from the SAGD process, either by numerical simulation or by an analytical approach.

CHAPTER 3

PROBLEMS AND OBJECTIVES

As stated in Chapter 1, the extension of the conventional Darcy's equation to multiphase flow does not give an accurate prediction of relative permeability curves, which are fundamental to reservoir engineering. This is attributed to the presence of interfacial coupling effects. This research aims to gain a better understanding of such effects and to devise means by which such effects can be incorporated into more refined multiphase equations.

This is to be achieved through a combination of mathematical formulation, experimental data analysis and numerical simulation studies. For data analysis, some experimental data available in the literature and additional experimental data to be acquired in the laboratory are to be used.

Acquiring these data entails making some changes in the design of the available laboratory equipment and experimental technique. That is, a slight modification of some of the components of the equipment and experimental technique are required in order to be able to achieve the desired steady state and unsteady state experiments.

The equipment for conducting experiments was developed by Ayub (2000). Ayub also presented detailed descriptions of the equipment. The schematic representation of the equipment (or the experimental set up) is shown in Figure 3.1. The equipment consists of the core holder and injection caps, fluid injection system, effluent collection system, dynamic saturation measurement system, dynamic pressure measurement system and data acquisition systems.

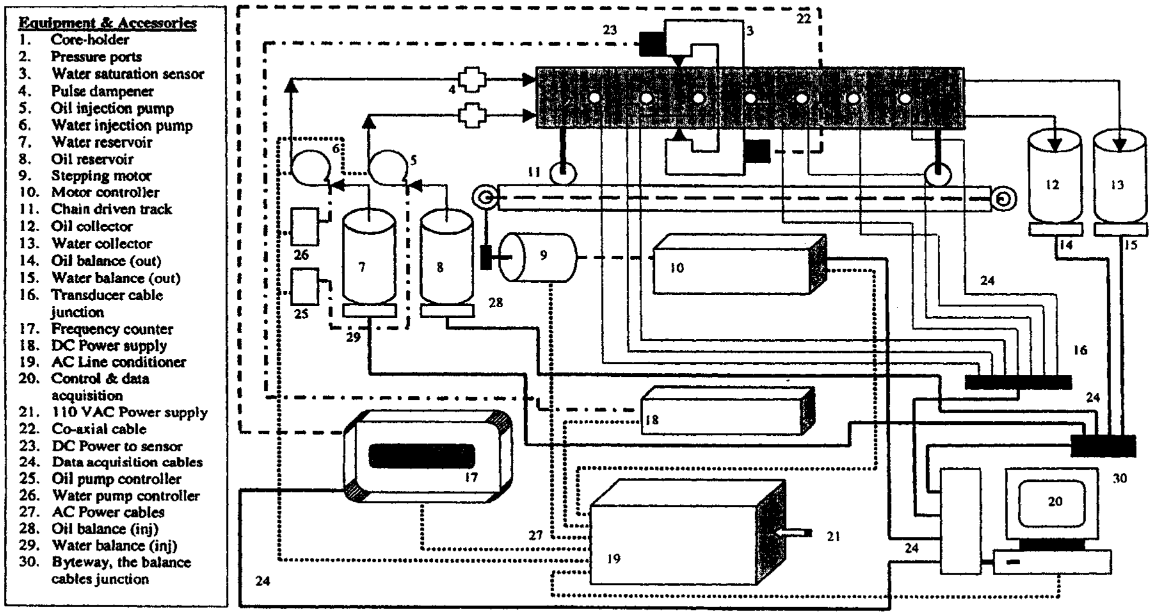


Figure 3.1: Schematic representation of equipment (experimental set up of steady and unsteady states two-phase flow)
Courtesy: Ayub (2000)

The required modifications involve improving the design of the end caps and the use of endplate surface membranes that enable the collection of the fluids separately, after flooding through the core. Also, the water-wet disc (plates) are to be treated with NAOH to enhance their affinity for the wetting phase. Tap water is to be used as the wetting phase instead of distilled water to enhance the anionic characteristics of the wetting phase.

In addition, the saturation measurement and pressure sensing system needed to be refined. The saturation measurements system is to be modified to take proper account of all the factors that contribute to the capacitance variations when measuring frequency. A new set of models is to be employed to derive saturation from frequency measurements. In the case of the pressure system, the possibility of using a new set of more sensitive sensors that reduce error as a percentage of total measurements is to be investigated, especially in the unsteady state experiments.

Additional work is to be carried out in the following order:

1) A detailed review of the modified set of transport equations in the previous work by Ayub (2000) with a view to identifying areas for improvements, if any. Possible areas of modification include the incorporation of momentum transfer (viscous coupling) and capillary coupling effects; and any other effect that comes into play as against only the capillary coupling effect, which was considered in Ayub's previous research study, into the formulation. In essence, the work involves modeling of 1-D, incompressible, two-phase flow through homogenous, isotropic porous media that incorporates gravity and capillary pressure, as well as both momentum transfer (viscous coupling) and capillary coupling effects and other effects such as hydrodynamic effects. Such equations can then be extended later to any type of dimension/geometry, any number or type of phases or any type of medium, once the theoretical formulations have been fully established and tested.
2) Development of a 1-D, two-phase numerical simulator based on the developed set of mathematical equations (transport equations) to predict pressure, saturation and production history. These numerical models would provide further validation of the theory (developed mathematical models). The simulation would enable a sensitivity analysis to be carried out. That is, it would enable quantification of the practical impact of not including interfacial momentum transfer (viscous coupling) and capillary coupling in two-phase flow. The simulator can then be extended to different reservoir situations such as 3-D and radial systems as well as to 3-phase once its applicability in 1-D, two-phase flow is demonstrated. The numerical simulation will entail the use of a finite difference scheme to find approximate solutions to the partial differential equations (derived through a combination of the modified Darcy's equations and the continuity equation), selection of appropriate grid systems and solving the resulting systems of equations. Also, recently introduced appropriate boundary conditions are to be incorporated into the simulator, instead of the generally assumed boundary conditions in the laboratory core flooding experiments.
3) Carry out experimental testing, with special emphasis on counter-current, steady state experiments. Even though unsteady state experiments might be employed in this study, more emphasis is not be placed on such experiments because of the problems associated with

pressure measurements in such experiments. Errors, as percentages of pressure measurements in such experiments, are always much higher.

4) Formulate analytical solutions that use data from experimental testing as input.
5) Compare the results from experimental testing with numerical simulation results and direct analytical solutions.
6) If the outcomes of the results show that there is need for modification of the mathematical models, this would be done and a new simulator would be developed and then compared again (see the flow diagram below for the stages involved).

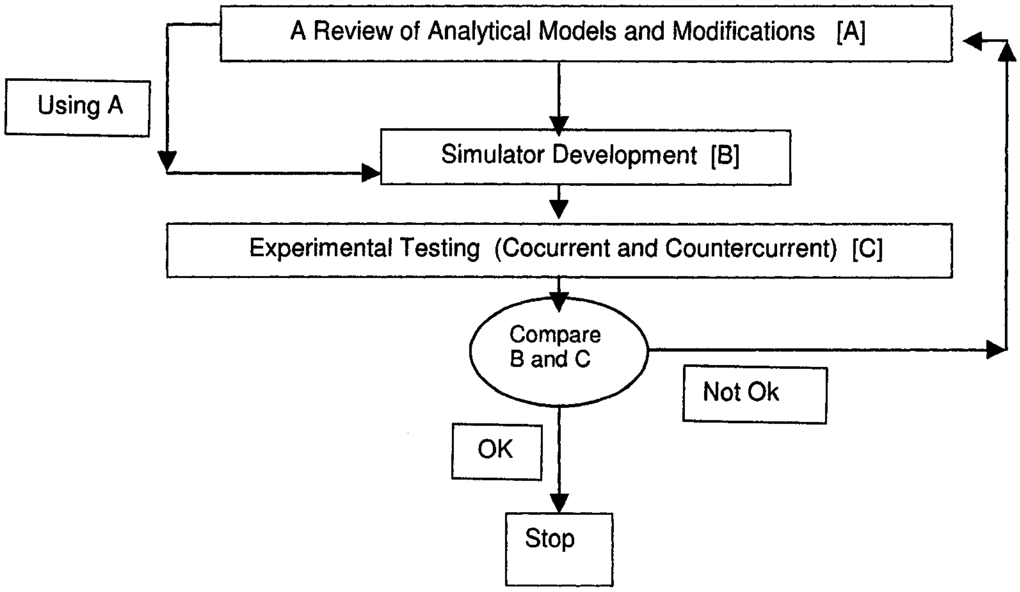

A Review of Analytical Models and Modifications [A]
Using A
Simulator Development [B]
Experimental Testing (Cocurrent and Countercurrent) [C]
Compare B and C
Not Ok
OK
Stop

Figure 3.2: Flow diagram of the stages involved in the research study.

Achieving the above set goals helps to quantify the impact of interfacial coupling in porous media, multiphase flow. This in turn helps to achieve better and more accurate prediction of hydrocarbon resources. Better and more accurate prediction of hydrocarbon resources leads to:

1) Enhancement of the design of surface facilities for predicted hydrocarbon production.
2) Improvement of the overall efficiency of drilling and production systems and reservoir forecasting, and hence, optimum allocation of financial resources to the development of drilling and production systems.

CHAPTER 4

MATHEMATICAL (TRANSPORT EQUATIONS') FORMULATIONS

4.1. Introduction

In this chapter, the principles and the concepts underlying the theory of the modified set of transport equations, which describe the simultaneous cocurrent or countercurrent flow of two phases through porous media, are presented. A modification of the pressure difference equation developed by Bentsen (1992), which forms the basis of the modified sets of equations, is presented. This is followed by a detailed derivation of the modified set of equations, and the defining equations for the viscous and capillary coupling parameters. The theory and principles presented here follow closely those developed by Bentsen (2001). Finally, the fractional flow equations, based on the modified transport equations, are presented.

One of the main objectives of presenting rigorous derivations in this chapter is to develop appropriate equations that accurately reflect the physics of the simultaneous flow of fluids in porous media. Based on the experimental results presented by Lelièvre (1966), Bourbiaux and Kalaydjian (1990), Kalaydjian (1990), Bentsen and Manai (1991 and 1993) and Bentsen (1998b), the magnitude of the relative permeabilities for a given phase and saturation obtained from a countercurrent flow experiment are always less than those obtained from a cocurrent flow experiment conducted in the same porous medium. There is no reason this should be so because the fluids flow in the same media even if the flow morphologies are different. Even if this is correct, there should be a relationship that accounts for this discrepancy such that, if countercurrent flow predominates in a given flow process, the equation should give the correct magnitude of the relative permeability for a given phase.

One explanation for the difference in magnitude of cocurrent and countercurrent relative permeabilities is that the equations that relate the relative permeabilities (and hence the mobilities) to other measured properties do not capture the correct physics of flow. This is because such equations do not include the effect of interfacial coupling between the two phases appropriately. Interfacial coupling has been identified in both cocurrent and countercurrent flow. It can be concluded from the experimental work of Lelièvre (1966) and Bentsen and Manai (1991 and 1993) that the interfacial coupling effects are significantly larger in a countercurrent flow experiment. When interfacial coupling effects arise in porous media, proper association of this effect with either the pressure/potential gradient or the mobility is required. Hence there is a need to resort to the more rigorous approach presented in this chapter.

Also, there is a need to present answers to the following questions when developing or presenting transport equations and finding a proper interpretation of the experiments conducted:

(1) Does cocurrent or countercurrent flow predominate in a given recovery process,

(2) Does the cocurrent or the countercurrent mobility (or relative permeability) represent the actual mobility of a given fluid,

(3) In the case where both countercurrent and cocurrent flow occur in a porous medium (spontaneous or free inhibition), how does one treat, or estimate, the mobilities,
(4) Which of the drainage and inhibition recovery processes occur in a given countercurrent or cocurrent flow,
(5) What is the order of magnitude of interfacial coupling in a cocurrent and a countercurrent flow experiment.

The above questions are pertinent because the usual transport equations for multiphase flow are defined for fluids flowing in the same direction (that is, cocurrent flow), and not for fluids flowing in opposite directions (that is, countercurrent flow). Hence, the use of mobilities that are determined for cocurrent flow, with the usual transport equation formulation, to predict a countercurrent flow process is questionable. The usual equations do not make any distinction between cocurrent and countercurrent flow. Hence, correction terms and/or coefficients, which do not exist in the usual transport equations, should appear in the transport equations to account for discrepancies in measured relative permeabilities determined in countercurrent and cocurrent flow experiments.

Countercurrent flow can either be gravity-driven or capillary-driven (pure countercurrent imbibition, also called piston-like imbibition) or a combination of both. An example of vertical gravity driven countercurrent flow is the upward movement of gas and the downward movement of oil that takes place when gas is liberated below the bubble point pressure in a solution gas drive or a depletion drive reservoir. This is really a very complex type of flow that is different from the one that takes place when a gas cap expands, which is basically a cocurrent type of flow. Relative permeability for this type of solution gas driven countercurrent flow is not defined (Chierici (1994)). Another type of countercurrent flow process is the upward movement of steam and the downward movement of bitumen and condensed steam (water) both at the edge of the steam chamber and partly at the top of the chamber in the steam-assisted gravity drainage (SAGD) process. Accurate relative permeability measurements for the SAGD process are also yet to be reported. The problem associated with this accurate measurement has been attributed to coupling (Nasr et al. (2000)). The SAGD process is a much more complex process with a more difficult recovery prediction pattern. A good relative permeability or mobility equation that incorporates the effect of interfacial coupling would improve the accuracy of the production prediction.

Also, countercurrent flow can be forced horizontal, without any gravity effect. Forced countercurrent flow rarely, if ever, occurs in the field, although this might be employed in laboratory experiments. The reason is that it makes the analysis easier and the results obtained from this type of flow can be applied easily to gravity driven and capillary driven counter current flow with a careful definition of capillary pressure as defined by Bentsen (2003a).

4.2. Pressure and Potential Difference Equation

When two fluids flow in a porous medium, in order to derive a more general relationship between the static and dynamic pressures, the pressure in a given phase within a porous medium may be divided, at a given time and position, into two terms. That is:

$$P_i = P_i^0 + P_i^1 \tag{4.1}$$

where P_i^0 is the pressure in phase i under conditions of static equilibrium, and P_i^1 is the small change in pressure in the same phase, that comes about as a consequence of dynamic action. From Equation 4.1, the difference in pressure between two flowing phases can be written as:

$$P_d = P_2 - P_1 = P_2^0 - P_1^0 + \left(P_2^1 - P_1^1\right) \tag{4.2}$$

The difference in pressure under static conditions, the static capillary pressure, P_c, is:

$$P_c = P_2^0 - P_1^0 \tag{4.3}$$

Equation 4.2 then becomes:

$$P_d = P_c + \left(P_2^1 - P_1^1\right) \tag{4.4}$$

From the experimental results of Bentsen and Manai (1991 and 1993), the connection between the gradients of the dynamic pressures in Equation 4.4 can be established by differentiating Equation 4.4 with respect to distance to give:

$$\frac{\partial P_d}{\partial x} = \frac{\partial P_c}{\partial x} + \left(\frac{\partial P_2^1}{\partial x} - \frac{\partial P_1^1}{\partial x}\right) \tag{4.5}$$

Rearranging Equation 4.5 and substituting the gradient of Equation 4.2 leads to:

$$\frac{\partial P_2^1}{\partial x} - \frac{\partial P_1^1}{\partial x} = \left(\frac{\partial P_2}{\partial x} - \frac{\partial P_1}{\partial x}\right) - \frac{\partial P_c}{\partial x} \tag{4.6}$$

To be able to make use of Equation 4.6, it is necessary that the static pressures in a capillary pressure experiment be distinguished from the dynamic pressures that are measured in an experiment that can be used to define the RHS of Equation 4.6. One way this can be achieved is to set $\frac{\partial P_c}{\partial x} = 0$ (that is, to conduct a steady-state experiment). Under steady state conditions, Equation 4.6 becomes:

$$\frac{\partial P_2^1}{\partial x} - \frac{\partial P_1^1}{\partial x} = \left(\frac{\partial P_2}{\partial x} - \frac{\partial P_1}{\partial x}\right) \tag{4.7}$$

Considering 3-dimensional flow in the i, j, k directions and expanding the right-hand-side of Equation 4.7, gives:

$$\nabla P_d = \nabla P_2^1 - \nabla P_1^1 = i\left(\frac{\partial P_2}{\partial x_1} - \frac{\partial P_1}{\partial x_1}\right) + j\left(\frac{\partial P_2}{\partial x_2} - \frac{\partial P_1}{\partial x_2}\right) + k\left(\frac{\partial P_2}{\partial x_3} - \frac{\partial P_1}{\partial x_3}\right) \tag{4.8}$$

Based on the experimental results of Bentsen and Manai (1991 and 1993):

$$\frac{\partial P_1}{\partial x_i} = R_{12} \frac{\partial P_2}{\partial x_i} \quad \text{, for i = 1,2,3} \tag{4.9}$$

where

$$R_{12} = 1 - a(1 - S) \tag{4.10}$$

and R_{12} is a weak function of normalized saturation that is introduced to account for the fact that, for vertical, steady-state cocurrent flow, the potential gradient for the wetting phase is not parallel to that for the nonwetting phase. The parameter a is a dimensionless parameter whose magnitude must be determined experimentally. It should be noted that when formulating a countercurrent flow equation, the sign of R_{12} changes to negative (Bentsen (1992,1994a, 1994b and1997)) because the pressure (or potential) gradients are acting in opposite directions. The incorporation of this sign effect into the general equation is shown later in the section where capillary coupling is discussed.

Substituting Equation 4.9 into Equation 4.8 yields:

$$\nabla P_d = \nabla P_2^1 - \nabla P_1^1 = (1 - R_{12})\left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] \ldots\ldots\ldots\ldots\ldots\ldots (4.11)$$

Now one needs to substitute the defining equation for the difference in dynamic pressure gradient back into the original equation. Substituting Equation 4.11 into the 3-dimensional form of Equation 4.5 leads to:

$$\nabla P_d = \nabla P_c + (1 - R_{12})\left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] \ldots\ldots\ldots\ldots\ldots\ldots (4.12)$$

Rearranging Equation 4.12 gives:

$$\nabla P_c = \nabla P_d - (1 - R_{12})\left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] \ldots\ldots\ldots\ldots\ldots\ldots (4.13)$$

Taking the gradient of Equation 4.2, one obtains:

$$\frac{\partial P_d}{\partial x} = \frac{\partial P_2}{\partial x} - \frac{\partial P_1}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.14)$$

The gradient in 3-dimensional notation can be written as:

$$\nabla P_d = \nabla P_2 - \nabla P_1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.15)$$

Introducing Equation 4.15 into Equation 4.13 leads to:

$$\nabla P_c = \left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] - \left[i\frac{\partial P_1}{\partial x_1} + j\frac{\partial P_1}{\partial x_2} + k\frac{\partial P_1}{\partial x_3} \right] + \ldots\ldots\ldots\ldots$$

$$+ (R_{12} - 1)\left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.16)$$

Rearranging Equation 4.16 leads to:

$$\nabla P_c = R_{12}\left[i\frac{\partial P_2}{\partial x_1} + j\frac{\partial P_2}{\partial x_2} + k\frac{\partial P_2}{\partial x_3} \right] - \left[i\frac{\partial P_1}{\partial x_1} + j\frac{\partial P_1}{\partial x_2} + k\frac{\partial P_1}{\partial x_3} \right] \ldots\ldots\ldots\ldots (4.17)$$

For one-dimensional flow, Equation 4.17 becomes:

$$\frac{\partial P_c}{\partial x} = R_{12}\frac{\partial P_2}{\partial x} - \frac{\partial P_1}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.18)$$

Finally, writing Equation 4.18 in terms of potentials results in:

$$\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g = \frac{\partial \psi_2}{\partial x} R_{12} - \frac{\partial \psi_1}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.19)$$

where

$$\frac{\partial \psi_i}{\partial x} = \rho_i g + \frac{\partial P_i}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.20)$$

Equations 4.19 and 4.20 are applicable to unsteady-state, vertical flow.

4.3. Derivation of Basic Equations

There is a need to find the equations that govern one-dimensional cocurrent and countercurrent flow. Such equations should also incorporate the effect of viscous and capillary coupling. The derivations of such equations are presented in this and subsequent sections.

As proposed by de Gennes (1983) and Kalaydjian (1987 and 1990), under conditions of vertical flow of two continuous phases in a porous medium, the equations of flow defining the fluxes of phases 1 and 2, respectively, in terms of the generalized phase mobilities are given as:

$$v_1 = -\lambda_{11} \frac{\partial \psi_1}{\partial x} - \lambda_{12} \frac{\partial \psi_2}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.21)$$

and

$$v_2 = -\lambda_{21} \frac{\partial \psi_1}{\partial x} - \lambda_{22} \frac{\partial \psi_2}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.22)$$

The meaning of all the terms in the equations above and below are given in the nomenclature. Also, Equations 4.21 and 4.22 are confined to the stable, collinear, vertical flow of two immiscible, incompressible fluids through a water-wet, isotropic and homogenous porous medium where phase 1 is the wetting phase and phase two is the non-wetting phase.

Substituting Equation 4.19 into Equations 4.21 and 4.22, respectively, gives:

$$v_1 = -\left(\lambda_{11} + \frac{\lambda_{12}}{R_{12}}\right)\frac{\partial \psi_1}{\partial x} - \frac{\lambda_{12}}{R_{12}}\left(\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right) \ldots\ldots\ldots\ldots(4.23)$$

and

$$v_2 = -(\lambda_{21} R_{12} + \lambda_{22})\frac{\partial \psi_2}{\partial x} + \lambda_{21}\left(\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right) \ldots\ldots\ldots\ldots(4.24)$$

The R_{12} term in Equations 4.23 and 4.24 is defined in Equation 4.10. In Equation 4.10, if $a = 0$, then $R_{12} = 1$; that is, the hydrodynamic effects are neglected (Bentsen (1998a and 1998b)).

The conventional transport equations, under the assumptions made above, may be written as:

$$v_1 = -\lambda_1 \frac{\partial \psi_1}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4.25)$$

$$v_2 = -\lambda_2 \frac{\partial \psi_2}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 4.26)$$

In horizontal, steady-state, cocurrent flow, the potential gradients, $\partial \psi_2 / \partial x$ and $\partial \psi_1 / \partial x$, act in the same direction. Thus, by comparing Equations 4.25 and 4.26 with Equations 4.23 and 4.24, it can be inferred that:

$$\left(\lambda_{11}+\frac{\lambda_{12}}{R_{12}}\right)=\lambda_1^0 \qquad (4.27)$$

and

$$\left(\lambda_{21}R_{12}+\lambda_{22}\right)=\lambda_2^0 \qquad (4.28)$$

where the λ_i^0, (i = 1,2), are the mobilities determined in a horizontal, steady state, cocurrent flow experiment.

In horizontal, countercurrent, steady-state flow, the potential gradients, $\partial\psi_2/\partial x$ and $\partial\psi_1/\partial x$, act in opposite directions. Hence, from Equations 4.25 and 4.26, and Equations 4.23 and 4.24, it follows that, for countercurrent flow:

$$\left(\lambda_{11}-\frac{\lambda_{12}}{R_{12}}\right)=\lambda_1^* \qquad (4.29)$$

and

$$\left(\lambda_{22}-\lambda_{21}R_{12}\right)=\lambda_2^* \qquad (4.30)$$

where the λ_i^*, (i = 1,2) , are the mobilities determined in a horizontal, steady-state, countercurrent flow experiment.

On the basis of experimental results presented in the literature (Lelièvre (1966); Bourbiaux and Kalaydjian (1990); Kalaydjian (1990); Bentsen and Manai (1991,1993), Naser et al. (2000)), it appears that mobilities determined in a countercurrent experiment are less than those determined for the same sand-fluid system, in a cocurrent experiment. Using the experimental results of Bentsen and Manai (1991 and 1993), it can be inferred that (Bentsen (1998b)):

$$\lambda_i^*=\alpha_i\,\lambda_i^0, \qquad i=1,\ 2 \qquad (4.31)$$

where the α_i, i = 1,2, are the parameters that control the amount of interfacial coupling that takes place.

On substituting Equation 4.31 into Equations 4.29 and 4.30, Equations 4.27, 4.28, and Equations 4.29 and 4.30 comprise a system of four equations involving six unknowns: $\lambda_{11},\lambda_{12},\lambda_{21},\lambda_{22},\alpha_1$ and α_2. This system of equations may be solved to arrive at:

$$\lambda_{11}=\left(\frac{1+\alpha_1}{2}\right)\lambda_1^0 \qquad (4.32)$$

$$\frac{\lambda_{12}}{R_{12}}=\left(\frac{1-\alpha_1}{2}\right)\lambda_1^0 \qquad (4.33)$$

$$\lambda_{22}=\left(\frac{1+\alpha_2}{2}\right)\lambda_2^0 \qquad (4.34)$$

and

$$R_{12}\lambda_{21} = \left(\frac{1-\alpha_2}{2}\right)\lambda_2^0 \quad \text{.....................(4.35)}$$

Substituting Equations 4.32 and 4.33 into Equation 4.23, and Equations 4.34 and 4.35 into Equation 4.24 leads to:

$$v_1 = -\lambda_1^0\left(\frac{\partial\psi_1}{\partial x} + \left(\frac{1-\alpha_1}{2}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right]\right) \quad \text{.....................(4.36)}$$

$$v_2 = -\lambda_2^0\left(\frac{\partial\psi_2}{\partial x} - \left(\frac{1-\alpha_2}{2R_{12}}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right]\right) \quad \text{.....................(4.37)}$$

Neglecting the hydrodynamic effects, which introduces only a relative error of about 1 %, (Bentsen (1998a and 1998b)), R_{12} then equals 1 in both Equations 4.36 and 4.37. Equations 4.36 and 4.37 represent the newly modified Darcy's equation (basic equations) for two-phase flow in porous media. The equations are applicable to both cocurrent and countercurrent, unsteady-state flow. If there is no interfacial coupling, $\alpha_1 = \alpha_2 = 1$, and the equations reduce to the conventional equations.

These equations are also similar and consistent with those developed by Ayub (2000). Ayub's derivations were in terms of pressure gradients while the derivations presented here are in terms of potential gradients. No expression was given for the viscous coupling in the transport equations that he presented and the viscous coupling and hydrodynamic effects are neglected in the final analysis and interpretation of results.

To make the equations more meaningful, a proper definition of capillary pressure under the new derivation approach needs to be undertaken. Also, a proper definition and analysis of the interfacial coupling coefficients (parameters) needs to be undertaken. These are presented in the subsequent sections. The analysis presented in the sections that follow essentially aims to make further improvements in the definition and analyses of capillary pressure and coupling coefficients as presented in previous work by Ayub (2000), Bentsen (2003c) and Ayub and Bentsen (2004).

4.4. Capillary Pressure Equations

The usual approach in reservoir engineering, when describing the dynamics of two immiscible phases flowing through a porous medium, is to assume that Leverett's supposition, that the difference in pressure between the two phases is dictated by capillary pressure, holds in all situations along the length of the porous medium.

Drainage or imbibition processes can occur in the recovery process depending on the stage of recovery (primary or secondary) under which production takes place. In the majority of the situations, imbibition, an increase in the wetting phase saturation, predominates except in high water coning, gas flooding and solution gas drive recovery processes where drainage, a decrease in the wetting phase saturation, predominates. As a result of this, in this research, the relative permeability measurement was undertaken using an inhibition process and the theory of capillary pressure presented is applicable to the imbibition process.

The hysteresis between the drainage and imbibition processes can be shown on the same chart, showing plots of capillary pressure versus wetting phase saturation. Whichever of the plots is applicable is normally used as the input in the reservoir simulation process. So if the defining equation for the capillary pressure curve is not well formulated it could have considerable effect on the outcome of the predicted recovery.

In has been shown recently, using a detailed mathematical analysis, that Leverett's supposition does not hold in all situations (Bentsen (2003a)). The variation in capillary pressure that occurs depends on the type of conditions, such as gravity driven or capillary driven (pure inhibition) or horizontal forced, cocurrent or countercurrent, under which the recovery takes place. The outcome of the analysis by Bentsen is used in the definition of capillary pressure in this research and it is presented below.

The capillary pressure, depending on the type of flowing process taking place, can be defined as follows (Bentsen (2003a)):

$$P_c = P_2 - P_1 \qquad (4.38a)$$

$$P_c = R_{12}P_2 - P_1 \qquad (4.38b)$$

$$P_c = R_{12}P_2(x) + P_1(x) - C^o \qquad (4.39)$$

$$C^o = P_1(0) + P_1(L) \qquad (4.40)$$

Equation 4.38a is applicable to all situations at the inlet and along the length of the core or reservoir for both steady state and unsteady conditions and unforced horizontal, vertical and inclined, cocurrent flow, provided the hydrodynamic effects are neglected. Equation 4.38b is applicable only along the length of the core or reservoir for both steady state and unsteady conditions and unforced horizontal, vertical and inclined, cocurrent flow. Equations 4.39 and 4.40 are applicable to forced, steady state, countercurrent flow. The later two equations are used when the countercurrent flow experiments are steady state, and force driven (which occurs mostly in horizontal systems). The derivations of the capillary equation defined by Equations 4.38b, 4.39 and 4.40 are presented in a recent technical note by Bentsen (2003a).

4.5. Interfacial Coupling

Two types of interfacial coupling can take place in a porous medium. These are viscous and capillary coupling (Babchin and Yuan (1997), Bentsen (2001 and 2003c), Langaas and Papatzacos (2001) and Ayub and Bentsen (2004)). In the derived transport Equations 4.36 and 4.37, the mobility and potential gradient appear as a product. So, the coupling associated with the mobility (viscous coupling) and capillary pressure (capillary coupling), should also appear as a product. Thus, it follows that the interfacial coupling parameters, α_i, i = 1,2, may be defined by:

$$\alpha_i = \alpha_{vi}\alpha_{ci}\text{, i = 1,2} \qquad (4.41)$$

where α_{vi} is the viscous coupling parameter and α_{ci} is the capillary coupling parameter. The defining equation for each of these types of interfacial coupling is constructed below.

4.5.1. Viscous Coupling

In earlier work, viscous coupling is usually associated with the cross terms (relative permeability terms) in the matrix formulation of two-phase flow (de la Cruz and Spanos (1983), Whitaker (1986) Kalaydjian and Legait (1987), Spanos et al. (1988)). Also in the present formulation viscous coupling is associated with the mobilities, λ_{12} and λ_{21} (Bentsen (2001 and 2003c) and Ayub and Bentsen (2004)). Based on Onsager's (1931a and 1931b) reciprocal relations, it can be assumed (Rose (1988), Kalaydjian (1990), Liang and Lohrenz (1994)) that:

$$\lambda_{12} = \lambda_{21} \quad \ldots\ldots (4.42)$$

The defining equations (Bentsen (2003c) and Ayub and Bentsen (2004)) for the viscous coupling can be constructed to satisfy Equation 4.42 as follows:

$$\alpha_{v1} = 1 - \frac{c}{R_{12}} \frac{\lambda_2^0}{\lambda_m^0} \quad \ldots\ldots (4.43)$$

$$\alpha_{v2} = 1 - c\, R_{12} \frac{\lambda_1^0}{\lambda_m^0} \quad \ldots\ldots (4.44)$$

It follows, in view of Equations 4.33 and 4.35, that:

$$\lambda_{12} = \lambda_{21} = \frac{c}{2} \cdot \frac{\lambda_1^0 \lambda_2^0}{\lambda_m^0} \quad \ldots\ldots (4.45)$$

where c is a parameter that controls the amount of viscous coupling and it can be determined experimentally. The parameter, λ_m^0, is introduced to ensure dimensional consistency. When $\lambda_m^0 = \lambda_{1r}^0$, the normalized saturation, S, equals 1; and when $\lambda_m^0 = \lambda_{2r}^0$, S equals 0. These conditions are satisfied by:

$$\lambda_m^0 = S\lambda_{1r}^0 + (1-S)\lambda_{2r}^0 \quad \ldots\ldots (4.46)$$

4.5.1.1 Quantification of Flow Due to Viscous Coupling

Combining Equations 4.19, 4.21, 4.33 (modified for horizontal flow) and Equations 4.36 and 4.43, it may be shown that, under steady-state conditions, the fraction of the total wetting-phase flux, v_1, that arises because of viscous coupling is defined by:

$$\frac{v_{12}}{v_1} = \frac{c}{2R_{12}} \frac{\lambda_2^0}{\lambda_m^0} \quad \ldots\ldots (4.47)$$

where v_{12} is the amount of wetting-phase flux due to viscous coupling. The maximum value of (v_{12}/v_1) occurs when $S = 0$; that is, $(\lambda_2^0/\lambda_m^0) = 1$, leading to:

$$\left(\frac{v_{12}}{v_1}\right)_{\max} = \frac{c}{2R_{12}(0)} \quad \text{......(4.48)}$$

From Equation 4.10, the term $R_{12}(0)$ is given as:

$$R_{12}(0) = 1 - a(1-0) = 1 - a \quad \text{......(4.49)}$$

Equation 4.48 then becomes:

$$\left(\frac{v_{21}}{v_1}\right)_{\max} = \frac{c}{2(1-a)} \quad \text{......(4.50)}$$

In a similar way, combining Equations 4.19, 4.22, 4.35 (modified for horizontal flow) and Equations 4.37 and 4.44, it may be shown that, under steady-state conditions, the fraction of the total non-wetting phase flux, v_2, that arises because of viscous coupling is defined by:

$$\frac{v_{12}}{v_1} = \frac{cR_{12}}{2}\frac{\lambda_1^0}{\lambda_m^0} \quad \text{......(4.51)}$$

where v_{21} is the amount of non-wetting phase flux due to viscous coupling. The maximum value of (v_{21}/v_2) occurs when $S = 1$; that is, $(\lambda_1^0/\lambda_m^0) = 1$, leading to:

$$\left(\frac{v_{21}}{v_2}\right)_{\max} = \frac{R_{12}(1)c}{2} \quad \text{......(4.52)}$$

From Equation 4.10, the term $R_{12}(1)$ is given as:

$$R_{12}(1) = 1 - a(1-1) = 1 - 0 = 1 \quad \text{......(4.53)}$$

Equation 4.52 then becomes:

$$\left(\frac{v_{12}}{v_1}\right)_{\max} = \frac{c}{2} \quad \text{......(4.54)}$$

Rakotomalala et al. (1995), showed that the fraction of the interfacial contact area of an idealized porous system roughly approximates the porosity of a natural porous system provided the porosity is low ($0 < \phi < 0.35$). Using lattice-gas simulations and analytical approaches, Rakotomalala et al. also showed that the interfacial contact area fraction, which controls the viscous coupling, is approximately equal to ϕ^2. The interfacial contact fraction (also called the viscous coupling function by Rakotomalala et al.) for a given phase, "provides a shear-induced flow due to the shear of the other phase" and "enhances the effect of the applied shear rate of its own" (Rakotomalala et al. (1995)). Hence, the fraction of the total flux that is due to viscous coupling in a given phase is approximately equal to ϕ^2.

For an average typical porosity value of 10% in natural porous media, this implies that the amount of the total flux due to viscous coupling in a given phase is insignificant (about 10^{-2} or less). This result agrees with that presented by Zarcone and Lenormand (1994) and Rakotomalala et al. (1995). Using experimental results from a mercury-water system, with an imposed condition of zero pressure gradients on one of the phases, Zarcone and Lenormand (1994) showed that the cross-term (a measure of viscous coupling) enhancement is only of order 10^{-2}.

In the formulation by Rakotomalala et al. (1995), the hydrodynamic effects were not accounted for. Hence, in order to avoid violating Equation 4.42, and in keeping with the above results and observations, it is postulated that Equation 4.50 and 4.54 may be written as:

$$\left(\frac{v_{12}}{v_1}\right)_{\max} = \frac{c_{\max 1}}{2(1-a)} \approx \frac{\phi^2}{1-a}, \quad \text{provided } 0 < \phi < 0.35 \ldots (4.55)$$

and

$$\left(\frac{v_{12}}{v_1}\right)_{\max} = \frac{c_{\max 2}}{2} \approx \phi^2, \quad \text{provided } 0 < \phi < 0.35 \ldots\ldots\ldots\ldots (4.56)$$

The condition, $0 < \phi < 0.35$, is practicable because the porosity of natural porous media falls in this range. If the subscripts 1 and 2 are introduced into $c_{\max}$ in Equations 4.54 and 4.55, to represent the wetting and the nonwetting phase respectively, and by rearranging the equations, leads to:

$$c_{\max 1} \approx 2\phi^2, \quad \text{provided } 0 < \phi < 0.35 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.57)$$

and

$$c_{\max 2} \approx 2\phi^2, \quad \text{provided } 0 < \phi < 0.35 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.58)$$

Substituting Equation 4.57 into Equation 4.43 at the maximum value of (v_{12}/v_1), that is, $(\lambda_2^0/\lambda_m^0) = 1$ and $R_{12} = 1 - a$, yields:

$$\alpha_{v1\max} = 1 - \frac{2\phi^2}{(1-a)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.59)$$

Similarly, substituting Equation 4.58 into Equation 4.44 at the maximum value of (v_{21}/v_2), that is, $(\lambda_1^0/\lambda_m^0) = 1$ and $R_{12} = 1$, yields:

$$\alpha_{v2\max} = 1 - 2\phi^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.60)$$

From the final definitions of α_{v1} and α_{v2}, it can be seen that the maximum values of α_{v1} and α_{v2} are approximately equal to 1; that is, the maximum values of the viscous coupling effect in both phases are negligible. The viscous effect varies throughout the saturation range with the maximum values quantified above occurring at the end points for each of the phases. If the hydrodynamic effects are neglected in Equations 4.43 and 4.44, that is $a =$ 0, then:

$$\alpha_{v1} = 1 - c_1 \frac{\lambda_2^0}{\lambda_m^0} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.61)$$

and

$$\alpha_{v2} = 1 - c_2 \frac{\lambda_1^0}{\lambda_m^0} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4.62)$$

Also, if the hydrodynamic effects are neglected, Equations 4.59 and 4.60 become:

$$\alpha_{v1\max} = \alpha_{v2\max} \approx 1-2\phi^2 \text{ ....(4.63)}$$

The values of c_1, c_2, $c_{\max 1}$ and $c_{\max 2}$ were investigated through experimental data analysis and numerical simulation as described in Chapter 7.

If the results above are true, then viscous coupling cannot be used to explain the experimental observations of various investigators (Lelièvre (1966), Bentsen and Manai (1991and 1993)). That is, for a given potential gradient, the total flux in a steady-state, countercurrent flow experiment is less than that in a steady state, cocurrent flow experiment. If such results hold, a different kind of interfacial coupling must be postulated. One option is to theorize that interfacial coupling takes place because of the difference in pressure (capillary pressure) that exists across the the fluid-fluid interfaces in a porous medium (Bentsen (2001)).

4.5.2. Capillary Coupling

Using channel flow theory (Rapport and Leas (1953) and Chatenever and Calhoun (1952)) and the concept of representative macroscopic surface (RMS), Ayub (2000), Bentsen (2003c) and Ayub and Bentsen (2004) constructed defining equations for the average (or effective) potential gradients for the wetting and the nonwetting phase.

Ayub (2000), Bentsen (2003c) and Ayub and Bentsen (2004) combined this equation with an expression for the capillary pressure gradient and other equations to show that another type of coupling called capillary coupling exists in porous media two-phase flow. The expression for capillary coupling, for vertical and horizontal flow, is given as the capillary coupling parameters, α_{c1} and α_{c2}:

$$\alpha_{c1} = \alpha_{c2} = \alpha_c = 1-\phi \text{...(4.64)}$$

4.5.2.1 Quantification of Flow Due to Capillary Coupling

Combining Equations 4.21, 4.22, 4.33, 4.35, 4.19 (modified for horizontal flow), 4.36, 4.37, 4.41 and 4.64, the fraction of the total flux, v_i, that arises because of capillary coupling is given as:

$$\frac{v_{12}}{v_1} = \frac{v_{21}}{v_2} = \frac{\phi}{2} \text{...(4.65)}$$

The terms, v_{12} and v_{21}, are, respectively, the amounts of phase 1 and phase 2 flux that arise because of capillary coupling. From Equation 4.65, it can be inferred that the fraction of the total flux that arises because of capillary coupling is (for a typical average porosity value of 0.1) of order 10^{-1}. Hence, compared with the order of viscous coupling (10^{-2}) presented by Zarcone and Lenormand (1994) and Rakotomalala et al. (1995), and the results in Equations 4.59 and 4.60, the contribution of capillary coupling to the total flux is an order of magnitude greater than that of viscous coupling.

4.5.3. Interfacial Coupling Parameters

Substituting Equations 4.61 and 4.64, and Equations 4.62 and 4.64 into Equation 4.41, the interfacial coupling parameters, α_i, i = 1,2, become:

$$\alpha_i = \alpha_{vi}\alpha_{ci}\text{, i = 1,2} \quad \text{..(4.41)}$$

$$\alpha_1 = \left[1 - \frac{c_1}{R_{12}}\frac{\lambda_2^0}{\lambda_m^0}\right].(1-\phi) \quad \text{..(4.66)}$$

$$\alpha_2 = \left[1 - c_2 R_{12}\frac{\lambda_1^0}{\lambda_m^0}\right].(1-\phi) \quad \text{..(4.67)}$$

If the viscous coupling is small enough to be neglected as shown by Zarcone and Lenormand (1994) and Rakotomalala et al. (1995), Equations 4.66 and 4.67 then become:

$$\alpha_1 = \alpha_2 = \alpha = 1 - \varphi \quad \text{..(4.68)}$$

4.6. Fractional Flow Equation

Rearranging Equations 4.36 and 4.37, respectively, yields:

$$\frac{\partial \psi_1}{\partial x} = -\frac{\lambda_1^0}{v_1} - \left(\frac{1-\alpha_1}{2}\right).\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right] \quad \text{...............(4.69)}$$

$$\frac{\partial \psi_2}{\partial x} = -\frac{\lambda_2^0}{v_2} + \left(\frac{1-\alpha_2}{2R_{12}}\right).\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right] \quad \text{.............(4.70)}$$

Subtracting Equation 4.69 from Equation 4.70 (that is, difference in potential gradient) leads to:

$$\frac{\partial \psi_2}{\partial x} - \frac{\partial \psi_1}{\partial x} = -\frac{R_{12}v_2}{\lambda_2^0} + \frac{v_1}{\lambda_1^0} + \left(1 - \frac{(\alpha_1 + \alpha_2)}{2}\right).\left[\frac{\partial P_C}{\partial x} - \Delta\rho g\right] .\text{(4.71)}$$

where

$$\Delta\rho = \rho_1 - R_{12}\rho_2 \quad \text{..(4.72)}$$

Combining Equations 4.19 and 4.71 and rearranging using the same approach by Bentsen (1998a and 1998b) gives:

$$f_1 = F_1\left[1 + \frac{(\alpha_1 + \alpha_2)}{2R_{12}}\frac{\lambda_2^0}{v}\left(\frac{\partial P_C}{\partial x} - \Delta\rho g\right)\right] \quad \text{.........................(4.73)}$$

where

$$F_1 = \frac{R_{12}\lambda_1^0}{R_{12}\lambda_1^0 + \lambda_2^0} \quad \text{..(4.74)}$$

and

$$v = v_1 + v_2 \quad \text{..(4.75)}$$

Equation 4.73 is similar to the Leverett fractional flow equation, except that Equation 4.73 incorporates the interfacial coupling effects. If there is no interfacial coupling, $\alpha_1 = \alpha_2 = 1$; hence, the interfacial coupling term disappears and the equation becomes:

$$f_1 = F_1\left[1 + \frac{\lambda_2^0}{v}\left(\frac{\partial P_C}{\partial x} - \Delta\rho g\right)\right] \text{.......................................(4.76)}$$

Also, if the amount of viscous coupling is negligibly small), Equation 4.73 becomes:

$$f_1 = F_1\left[1 + \frac{(1-\phi)\lambda_2^0}{v}\left(\frac{\partial P_C}{\partial x} - \Delta\rho g\right)\right] \text{..............................(4.77)}$$

CHAPTER 5

NUMERICAL SIMULATION BASED ON THE MODIFIED TRANSPORT EQUATIONS

5.1. Introduction

This chapter covers the development of a numerical simulator based exclusively on the modified set of mathematical models (transport equations) in Chapter 4. It covers the basic assumptions in the models, theoretical background, mathematical formulations, normalization, transformations between Eulerian and Lagrangian forms, discretization scheme, specification of boundary conditions, solution methods and codification (programming) using the Java^{TM} programming language.

5.2. Objectives

The development of the numerical simulator broadly follows the following steps:

1) Definition of the problem with partial differential equations and appropriate initial and boundary conditions.
2) Conversion of the partial differential equations to finite difference equations.
3) Applying the finite difference equations to the "discretized"- grid reservoirs.
4) Solving the equations to determine the behaviour of the reservoir; that is, determining the fractional flow (production), pressure and saturation profiles.

The derivation of the governing partial differential equations, from the transport equations derived in Chapter 4 and the continuity equations is presented, followed by the steps stated above. It should be noted that the mathematical formulations (transport equations) in Chapter 4 are a modified form of Darcy's equation that incorporates the effects of interfacial coupling in both cocurrent and counter current flow for steady and unsteady flow situations.

5.3. Theoretical Considerations and Background Information

Construction of numerical solutions to the modified transport equations in Chapter 4 and the associated equations can be accomplished in several ways. Whatever method is selected, a numerical solution generally involves combining the transport equations into manageable forms and writing numerical approximations of the resulting equations. This is followed by the setting-up of the system of algebraic equations for solution using a digital computer.

In one-dimensional flow problems, fractional flow can be used. The technique developed by Blair (Collins (1961)) for the pure inhibition process and Douglas et al. (1959) for multi-dimensional flow can be used also, depending on which is more convenient. The advantage of using fractional flow is that it can be converted easily to dimensionless form, thus reducing the number of variables that need to be handled. Also, the mathematical equations describing the flow process can be reduced to a single equation leading to one equation for every block.

In this research, the fractional flow concept and the technique introduced by Buckley and Leverett (1942) are used in combining the equations into a manageable form. Several authors have demonstrated the appropriateness of the fractional flow technique for one-dimensional immiscible displacement (Buckley and Leverett (1942), Douglas et al. (1958), McEwen (1959), Fayers and Sheldon (1959), Hovanessian and Fayers (1961), Collins (1961), Bentsen (1976 and 1978)). These authors employed several approaches such as neglecting both gravity and capillary effects, neglecting only the gravity effect, neglecting only the capillary effect or considering both gravity and capillary effects. All their investigations involved a numerical solution.

Douglas et al. (1958) presented an Eulerian technique for solving linear water flooding problems, which involved using a transformed saturation variable and the use of a small mobile water saturation ahead of the flood front to enable the base of the floodfront to move forward. Fayers and Sheldon (1959) obtained solutions to the one-dimensional displacement equation using both the Lagrangian and the Eulerian approaches. McEwen (1959) used the method of characteristics for numerical solution of the linear displacement equation with capillary pressure. Hovanessian and Fayers (1961) also solved numerically the equations describing waterflooding experiments that include both gravity and capillary effects. Their method is similar to that of Douglas et al. (1958) because it involves the use of the Eulerian form of the fractional flow equation and the transformation of the resulting second-order nonlinear partial differential equations. Hovanessian and Fayers' solution differs from those of Douglas et al. because it includes the effect of gravity, allows calculation of pressure profiles and makes use of a tabular format for the input of relative permeability and capillary pressure values instead of the use of polynomial functions.

Bentsen (1978) developed a Lagrangian technique for solving the equations that result from the simultaneous flow of two immiscible fluids in porous media. His formulation assumes a homogeneous and isotropic medium and one-dimensional, incompressible flow. Bentsen's technique makes use of full normalization of all variables. Bentsen's Lagrangian equation for fractional flow is a very useful analysis tool because it incorporates all the variables in the displacement process including all time dependent (transient) and steady state problems. He solved the Lagrangian equation using an explicit finite-difference scheme. His Lagrangian approach used a transformation process to eliminate the dimensionless distance variable from the resulting equation, enabling the calculation of fractional flow or volume of fluid produced and injected as a function of time and saturation. He used the boundary conditions, $(f_1(0,\tau)=0,\ \ f_1(1,\tau)=1)$, but his initial condition $(S(0,\tau)=1)$ is not very accurate (Shen and Ruth, 1994a, 1994b and 1996). Bentsen observed that $0<S(0,\tau)<1$, but no material balance equation was presented to account for this. However, he used the material balance concept to show that $f_1(S^*,\tau)=1$ for $0<S^*<1$. The elapsed time required to achieve a given saturation profile can be determined using Bentsen's method. Saeedi (1979) and Bentsen and Saeedi (1981) also presented numerical results obtained by solving the Lagrangian formulation of the immiscible displacement equation using the explicit finite difference technique (similar to that presented by Bentsen (1978)) and they compared the results with those obtained experimentally in the laboratory.

Shen and Ruth (1994a, 1994b and 1996) numerically solved the transient and the steady state forms of Bentsen's equation using the Galerkin-finite element method. They also formulated a new set of initial conditions for Bentsen's equation. They found that Bentsen's initial condition is unsatisfactory because the solutions obtained using his conditions did not satisfy the governing equations. Shen and Ruth formulated initial and boundary conditions, which satisfy the governing equations and are physically and mathematically consistent. They also showed that, normalized saturation rises gradually from 0 to 100% and hence it is time dependent.

It should be noted that the transport equations used in the numerical formulation in this research are different from those used in the above cited simulation studies and generally used in petroleum engineering and porous media studies. The derivation of the transport equations used in the numerical simulation process that follows is presented in Chapter 4.

5.4. Mathematical Description

The choice of numerical simulation as an analysis tool in this research is based on the fact that it enables sensitivity analysis to be carried out on the newly developed transport equations. Hence, the effect of neglecting either viscous or capillary coupling can be simulated easily. Also the accuracy of the new transport equations in predicting core-flooding laboratory experimental results can be verified easily.

Numerical simulation of core-flooding experiments is routinely employed in the study of immiscible displacement processes to analyze and interpret the results of unsteady state experiments such as history matching to obtain relative permeability and capillary pressure curves. Examples of such studies include those by Archer and Wong (1973), Sigmund and McCaffery (1979), Chavent et al. (1980), Batycky et al. (1981), Kerig and Watson (1987), Fassihi (1989), and Maini et al. (1990). The use of numerical simulation to test the modified set of transport equations through the use of an algorithm based on the new equations is another way in which numerical simulation can be employed. This method is better because it differs from the conventional history matching of field production in that the laboratory condition under which the experiment is conducted is controllable and can be built accurately into the simulator. The descriptions of the transport equations used in the simulation process are given in this section.

5.4.1. Basic Equations

Now let us consider the transport equations derived in Chapter 4. The modified set of transport equations (Equations 4.36 and 4.37) developed in Chapter 4 for one-dimensional flow, which incorporates viscous coupling, capillary coupling and hydrodynamic effects, is given as:

$$v_1 = -\lambda_1^0\left(\frac{\partial\psi_1}{\partial x} + \left(\frac{1-\alpha_1}{2}\right).\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right]\right) \text{........(5.1a)}$$

and

$$v_2 = -\lambda_2^0\left(\frac{\partial\psi_2}{\partial x} - \left(\frac{1-\alpha_2}{2R_{12}}\right).\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g\right]\right) \text{.......(5.1b)}$$

The definitions of all the terms in Equations (5.1a) and (5.1b) are given in the nomenclature. Neglecting the hydrodynamic effects makes R_{12} equal to 1 in both Equations (5.1a) and (5.1b). The equations are applicable to both cocurrent and countercurrent, unsteady state flow. If there is no interfacial coupling, $\alpha_1 = \alpha_2 = 1$, and the equations reduce to the conventional Darcy's equations (applied to multiphase flow). Basic assumptions in the derivation of the above equations are: 1) incompressible flow, 2) stable flow, 3) one-dimensional flow, 4) collinear flow, 5) horizontal, vertical or inclined flow of two immiscible and incompressible fluids, 6) flow is through a water-wet, isotropic and homogenous porous medium, 7) effects of gravity and capillarity are included, 8) phase 1 is the wetting-phase and phase 2 is the non-wetting phase.

By coupling Equation (5.1a) with Equation (5.1b), the fractional flow of the wetting phase can be written as:

$$f_1 = F_1\left[1 + \frac{(\alpha_1 + \alpha_2)}{2R_{12}}\frac{\lambda_2^0}{v}\left(\frac{\partial P_c}{\partial x} - \Delta\rho g \sin\Theta\right)\right] \quad \text{.................(5.1c)}$$

By putting $\bar{\lambda}(S) = (\alpha_1 + \alpha_2)/2$, Equation (5.1c) becomes:

$$f_1 = F_1\left(1 - \frac{\bar{\lambda}\lambda_2^0 \Delta\rho g \sin\Theta}{vR_{12}} + \frac{\bar{\lambda}\lambda_2^0}{vR_{12}}\frac{\partial P_c}{\partial x}\right) \quad \text{..........................(5.1d)}$$

where

$$F_1 = \frac{R_{12} M_r K_{r1}^0}{R_{12} M_r K_{r1}^0 + K_{r2}^0} \quad \text{..(5.2a)}$$

and

$$\Delta\rho = \rho_1 - R_{12}\rho_2 \quad \text{..(5.2b)}$$

The angle of inclination of the reservoir is introduced into Equation (5.1c) to account for the fact that the reservoir or the core can be inclined, horizontal or vertical. For a horizontal system Θ = 0 and sin Θ = 0, making the third term in the RHS of Equation (5.1c) equal to zero. For a vertical system, Θ = 90 and sin Θ = 1. Other equations, given in Chapter 4, needed to describe the flow of two phases flowing simultaneously through a porous medium are:

$$\lambda_i^* = \alpha_i \lambda_i^0, \quad i = 1,2 \quad \text{...(5.3)}$$

$$\alpha_i = \alpha_{vi}\alpha_{ci}, \quad i = 1,2 \quad \text{...(5.4)}$$

$$\alpha_{c1} = \alpha_{c2} = 1 - \phi \quad \text{...(5.5)}$$

$$\alpha_{V1} = 1 - \frac{c_1}{R_{12}}\frac{\lambda_2^0}{\lambda_m^0}, \quad \text{maximum value of } c_1 = c_{\max 1} = 2\phi^2 \quad \text{.......(5.6)}$$

$$\alpha_{V2} = 1 - c_2 R_{12} \frac{\lambda_1^0}{\lambda_m^0}, \text{ maximum value of } c_2 = c_{\max 2} = 2\phi^2 \ldots(5.7)$$

$$\lambda_m^0 = S\lambda_{1r}^0 + (1-S)\lambda_{2r}^0 \ldots\ldots(5.8)$$

$$P_c = P_2 - P_1 \ldots\ldots(5.9a)$$

$$P_c = R_{12}P_2 - P_1 \ldots\ldots(5.9b)$$

$$P_c = R_{12}P_2(x) + P_1(x) - C^o \ldots\ldots(5.9c)$$

$$C^o = P_1(0) + P_1(L) \ldots\ldots(5.9d)$$

$$R_{12} = 1 - a(1-S) \ldots\ldots(5.9e)$$

Equation (5.9a) is applicable to all situations at the inlet and along the length of the core or reservoir for both steady state and unsteady conditions and unforced horizontal, vertical and inclined, cocurrent flow, provided the hydrodynamic effects are neglected. Equation (5.9b) is applicable only along the length of the core or reservoir for both steady state and unsteady conditions and unforced horizontal, vertical and inclined, cocurrent flow. Equations (5.9c) and (5.9d) are applicable to forced, steady state, countercurrent flow. The latter two equations are used when the countercurrent flow experiments are steady state, and force driven (which occurs mostly in horizontal systems). Bentsen (2003a) presented the derivation of the capillary equation defined by Equations (5.9b), (5.9c) and (5.9d) in a recent technical note. The S in Equation (5.8) is the normalized saturation, the definition of which is given in the nomenclature and in Equation (5.31).

5.4.2. Differential Equations

The continuity equation for two-phase, immiscible, incompressible flow in homogenous and isotropic porous media is given as:

$$\frac{\partial f_i}{\partial x} = -\frac{\phi}{v}\frac{\partial S_i}{\partial t}, \quad \text{i = 1, 2} \ldots\ldots(5.10)$$

or

$$\frac{\partial v_i}{\partial x} = -\phi\frac{\partial S_i}{\partial t}, \quad \text{i = 1, 2} \ldots\ldots(5.11)$$

Saturation is a function of time and distance; therefore, the derivative of saturation with respect to time can be written as:

$$\frac{dS_i}{dt} = \frac{\partial S_i}{\partial x}\frac{dx}{dt} + \frac{\partial S_i}{\partial t} \ldots\ldots(5.12)$$

If S_i is chosen to correspond with a fixed S_i, then the LHS of Equation (5.12) becomes:

$$\frac{dS_i}{dt} = 0 \ldots\ldots(5.13)$$

Equation (5.12) can then be written as:

$$-\frac{\partial S_i}{\partial t}=\left(\frac{\mathrm{d}x}{\mathrm{d}t}\right).\left(\frac{\partial S_i}{\partial x}\right)\text{...}(5.14)$$

Substituting Equation (5.13) into Equations (5.10) and (5.11) gives:

$$\frac{\partial x}{\partial t}=\frac{v}{\phi}\frac{\partial f_i}{\partial S_i},\quad \mathrm{i}=1{,}2\text{..}(5.15)$$

Equation (5.15) defines the frontal advancement or displacement equations for the wetting and non-wetting phases. This equation and Equations (5.3) to (5.9) completely define the relationship between the countercurrent and cocurrent flow processes.

5.5. Normalization of Fractional Flow, Displacement and Potential Equations

For convenience sake and in order to be consistent with the normalized saturation in Equation (5.8), it is usual to normalize the variables in the unsteady state form of the fractional flow equation (Equation (5.1d)) and the displacement equation (Equation (5.15)). Normalization provides a great deal of insight into the effect of the various parameters. Normalization is usually done in the manner of Handy and Hardley (1956), Douglas et al. (1958), Fayers and Sheldon (1959) and Bentsen (1976 and 1978). The normalized equations are as follows:

$$\frac{\partial f_1}{\partial S}=\frac{\partial \xi}{\partial \tau}\text{..}(5.16)$$

$$f_1(S,\tau)=G(S)-N_cC(S)\frac{\partial S}{\partial \xi}\text{..}(5.17)$$

$$\frac{\mathrm{d}}{\mathrm{d}S}\left[G(S)-N_cC(S)\frac{\partial S}{\partial \xi}\right]\frac{\partial S}{\partial \xi}=\frac{\partial S}{\partial \tau}\text{..................................}(5.18)$$

where

$$G(S)=\left(1-\frac{\bar{\lambda}(S)[1+(\rho_2/\Delta\rho')(1-R_{12})]N_gK_{r2}^0(S)}{R_{12}(S)M_r}\right)F_1(S)\,.\,(5.19)$$

$$C(S)=-\frac{1}{M_rR_{12}(S)}\bar{\lambda}(S)F_1(S)K_{r2}^0(S)\frac{\mathrm{d}\pi_c}{\mathrm{d}S}\text{......................}(5.20)$$

$$F_1(S)=\frac{R_{12}(S)M_rK_{r1}^0(S)}{R_{12}(S)M_rK_{r1}^0(S)+K_{r2}^0(S)}\text{..............................}(5.21)$$

$$N_g=\frac{\lambda_{1r}^0\Delta\rho'\,g\sin\Theta}{v}\text{..}(5.22)$$

$$\Delta\rho'=\rho_1-\rho_2\text{..}(5.23)$$

$$\lambda_{r1}^0(S)=\frac{\lambda_1^0(S)}{\lambda_{1r}^0}\text{..}(5.24)$$

$$\lambda_{r2}^{0}(S)=\frac{\lambda_{2}^{0}(S)}{\lambda_{2r}^{0}} \quad\text{..........(5.25)}$$

$$\pi_{c}(S)=\frac{P_c-P_d\,'}{A_c} \quad\text{..........(5.26)}$$

$$\tau=\frac{vt}{\phi L(1-S_{2r}-S_{1i})} \quad\text{..........(5.27)}$$

$$\xi=\frac{x}{L} \quad\text{..........(5.28)}$$

$$M_{r}=\frac{k_{1r}}{\mu_{1}}.\frac{\mu_{2}}{k_{2r}}=\frac{\lambda_{1r}^{0}}{\lambda_{2r}^{0}} \quad\text{..........(5.29)}$$

$$N_{c}=\frac{A_{c}\lambda_{1r}^{0}}{vL} \quad\text{..........(5.30)}$$

$$S=\frac{S_{1}-S_{1i}}{1-S_{2r}-S_{1i}} \quad\text{..........(5.31)}$$

The normalized Equations (5.16) to (5.18) are similar to those developed by Bentsen (1976 and 1978) with a few exceptions that arise due to the fact that Darcy's equation has been modified. Under the present normalization, the terms, $G(S)$, $N_c, C(S)$, $F_1(S)$, M_r and N_g, are partly defined in terms of the mobilities and not the relative permeabilities. The term $G(S)$ is partly defined in terms of the ratio of nonwetting density to the difference in densities. Also, the term $F_1(S)$ is defined partly in terms of $R_{12}(S)$, while the terms $G(S)$ and $C(S)$ are defined partly in terms of the average coupling coefficient, $\bar{\lambda}(S)$ and $R_{12}(S)$. The last exception is a major difference between the present normalization and previous efforts in this area.

Similarly, for simplicity's sake, there is a need to normalize the potential (or pressure) equation in order to find expressions for the potential or pressure distribution across the core or reservoir. The total potential drop, ψ_t, across the core is the sum of three potential drops. These are the potential drop behind the front where only the wetting phase flows, the potential drop where a mixture of both nonwetting and wetting phase flows and the potential drop beyond the front where only the nonwetting phase flows. Mathematically:

$$\psi_{t}=\psi_{1b}+\psi_{bf}+\psi_{2f} \quad\text{..........(5.32)}$$

Taking the gradient of Equation (5.32) leads to:

$$\frac{\partial\psi_{t}}{\partial x}=\frac{\partial\psi_{1b}}{\partial x}+\frac{\partial\psi_{bf}}{\partial x}+\frac{\partial\psi_{2f}}{\partial x} \quad\text{..........(5.33)}$$

Equations (5.1a) and (5.1b) can be rearranged to obtain:

$$\frac{\partial\psi_{1}}{\partial x}=-\frac{v_{1}}{\lambda_{1}^{0}}-\left(\frac{1-\alpha_{1}}{2}\right)\left[\frac{\partial P_{c}}{\partial x}-\Delta\rho g\sin\Theta\right] \quad\text{..........(5.34)}$$

$$\frac{\partial\psi_2}{\partial x}=-\frac{v_2}{\lambda_2^0}+\left(\frac{1-\alpha_2}{2R_{12}}\right)\left[\frac{\partial P_c}{\partial x}-\Delta\rho g\sin\Theta\right]\text{......................}(5.35)$$

Behind the front, only the wetting phase flows (the residual oil does not flow); and ahead of the front, only the nonwetting phase flows (the irreducible water does not flow). Hence, the fractional flow of each phase in these regions is either 0 or 1 (that is, $v_1 = v$, behind the front and $v_2 = v$, ahead of the front) and the interfacial coupling parameter, α_i, is 1. These conditions reduce Equations (5.34) and (5.35) to the conventional Darcy's equation and when normalized yield:

$$\frac{\mathrm{d}\pi_{\psi 1b}}{\mathrm{d}\xi(S=1,\tau)}=-\frac{1}{N_c}\quad\text{for}\quad 0<\xi<\xi_b\text{..........................}(5.36)$$

and

$$\frac{\mathrm{d}\pi_{\psi 2f}}{\mathrm{d}\xi(S=0,\tau)}=-\frac{M_r}{N_c}\quad\text{for}\quad \xi_f<\xi<1\text{..........................}(5.37)$$

provided the pressure or potential at the outlet end is zero. The terms M_r and N_c are defined by Equations (5.29) and (5.30), respectively. The definition of all the other terms and parameters are given in the nomenclature. In the region where both the wetting and the nonwetting phases are flowing, fractional flow comes into play. Either of Equations (5.34) and (5.35) gives the potential gradient in the wetting and nonwetting phase in the two-phase region because they are linked through the fractional flow equation. Hence, it is sufficient to use either of them to estimate the potential drop across the core or the reservoir. Here, the nonwetting phase equation (Equation (5.35)) is selected. By introducing fractional flow, Equation (5.35) becomes:

$$\frac{\partial\psi_2}{\partial x}=-\frac{vf_2}{\lambda_2^0}+\left(\frac{1-\alpha_2}{2R_{12}}\right)\left[\frac{\partial P_c}{\partial x}-\Delta\rho g\sin\Theta\right]\text{....................}(5.38)$$

Normalizing Equation (5.38) gives:

$$\frac{\partial\pi_{\psi 2}}{\partial\xi}=-\frac{f_2 M_r}{L\lambda_{r2}^0 N_c}+N^{o2}(S)\frac{\partial S}{\partial\xi}-\frac{N_g}{LN_c}\left(\frac{1-\alpha_2(S)}{2R_{12}}\right)\text{...}(5.39)$$

where

$$N^{o2}(S)=\frac{1-\alpha_2(S)}{2}\frac{\mathrm{d}\pi_c}{\mathrm{d}S}\text{..}(5.40)$$

Substituting Equations (5.36), (5.37) and (5.39) into the normalized form of the total potential gradient drop (Equation (5.33)) across the core leads to:

$$\frac{\partial\pi_{\psi t}}{\partial\xi}=\frac{M_r}{N_c}\left[-\frac{1}{M_r}-1-\left(\frac{f_2}{L\lambda_{r2}^0}-\frac{N_c N^{o2}(S)}{M_r}\frac{\partial S}{\partial\xi}+\frac{N_g}{LM_r}\left(\frac{1-\alpha_2(S)}{2R_{12}}\right)\right)\right].(5.41)$$

The first term in Equation (5.41) is applicable over the interval, $0<\xi<\xi_b$, and the second term is applicable over the interval $\xi_f<\xi<1$. On the other hand, the third group of terms is applicable over the interval $\xi_b<\xi<\xi_f$ and the saturation values that correspond to these intervals.

5.6. Lagrangian Transformation of Fractional Flow and Displacement Equations

In order to solve completely Equations (5.16) and (5.17), together with Equations (5.2) to (5.9) as functions of time, space and saturation, it is necessary to write and solve the Lagrangian configurations of the equations. The Eulerian configuration enables the equation to be solved as a function of time and space while the Lagrangian configuration allows the equation to be solved as a function of saturation and time. However, using the Lagrangian formulation does not provide a closed-loop solution. In the present work, the Lagrangian technique is employed. This is the same as the Lagrangian formulation developed by Bentsen (1978) and re-formulated by Shen and Ruth (1994a and 1994b). This method is adopted because it enables the prediction of the elapsed time necessary to obtain a particular saturation solution. Under certain circumstances, this method also fully or partially eliminates the need for the scaling requirement that the permeability or mobility curves in the model (laboratory core) and prototype (actual field reservoir) be the same. The transformation below is the same as that of Shen and Ruth (1994a and 1994b).

Differentiating Equation (5.16) with respect to S gives:

$$\frac{\partial^2 f_1}{\partial S^2}=\frac{\partial^2 \xi}{\partial \tau \partial S} \text{..(5.42)}$$

Re-arranging Equation (5.17) gives:

$$\frac{\partial \xi}{\partial S}=\frac{-N_c C(S)}{f_1(S,\tau)-G(S)} \text{...(5.43)}$$

Differentiating Equation (5.43) with respect to τ gives:

$$\frac{\partial^2 \xi}{\partial S \partial \tau}=\frac{N_c C(S)}{(f_1(S,\tau)-G(S))^2}\frac{\partial f_1}{\partial \tau} \text{.......................................(5.44)}$$

Equating Equations (5.42) and (5.44) and re-arranging yields:

$$(f_1(S,\tau)-G(S))^2\frac{\partial^2 f_1}{\partial S^2}=N_c C(S)\frac{\partial f_1}{\partial \tau} \text{..........................(5.45)}$$

Equation (5.45) is a second order, parabolic, partial differential equation. Equation (5.45) is a modified form of the Bentsen Equation (Shen and Ruth (1994a and 1994b)). Bentsen (1976 and 1978) first derived an equation of this form, which was subsequently re-derived by Shen and Ruth using the approach above. The difference between Equation (5.45) and the original form developed by Bentsen is in the definitions of some of the normalized terms. The definition of N_c, $C(S)$, $F_1(S)$, M_r and N_g given in Section 5.5 are different from the original definition of these terms as given by Bentsen (1976) because of the incorporation of interfacial coupling terms. Henceforth, until the end of this Chapter, the 1 and 2 subscripts are dropped and anywhere f, S, v, P, ψ or any other variable pertaining to the phases are written, it shall be assumed that the variables are applicable to the wetting phase, phase 1, except when otherwise stated.

5.7. Initial and Boundary Conditions

Appropriate boundary and initial conditions are required to describe core-flooding experiments and in order to be able to solve the partial differential equations in the previous section. In core flooding experiments, capillary end effects, both at the inlet and at the outlet ends, exist, and they greatly affect saturation and hence the pressure (or potential) distributions in the displacement process. These effects affect unsteady experiments much more than steady state experiments and they might even be neglected under steady state conditions.

To be able to account for these effects, a proper definition of saturation and the dependence of pressure on saturation need to be undertaken. These effects are sometimes neglected in the formulation of boundary conditions. Neglecting these effects leads to an incorrect definition of the boundary conditions. Recent work in this area, (Fassihi (1989), Shen and Ruth (1994a, 1994b, and 1996)), has recognized this and some improvements have been made in the definition of outlet-end and inlet-end saturation (Shen and Ruth (1994a, 1994b and 1996)). Using the fractional flow concept, it was usual to assume that the inlet saturation rises immediately to a maximum value 1 at a slow rate. Fayers and Sheldon (1959) and several investigators that employed the fractional flow concept used this boundary condition. It has, however, been shown over the past four decades that this is wrong, except in situations when the measurement equipment is designed to account for this end effect. The inlet saturation rises to one over time and it is time dependent. Subsequent investigators have modified this earlier approach. All relevant boundary and initial conditions are presented below. Accounting for the inlet saturation effect, Shen and Ruth (1994a, 1994b and 1996) derived the following generally accepted set of boundary and initial conditions for coreflooding experiment:

$$S^*(0)=0 \dots\dots \tag{5.46}$$

$$\xi(S^*,\tau)=0 \dots\dots \tag{5.47}$$

$$f(0,\tau)=0 \dots\dots \tag{5.48}$$

$$f(S^*,\tau)=1 \dots\dots \tag{5.49}$$

The parameter S^* is the normalized inlet saturation. "The variation of S^* is restrained" (Shen and Ruth (1996)) by the material balance equation that follows:

$$\int_0^{S^*}\xi(S,\tau)dS=\int_0^{S^*}\int_S^{S^*}\frac{N_c C(S)}{f(S,\tau)-G(S)}\mathrm{d}S\mathrm{d}S=\tau \dots\dots \tag{5.50}$$

These initial and boundary conditions are adopted here because they have been shown to be more consistent and accurate in the simulation of core flooding experiments, as in the present work.

The parameter, S^*, can be computed iteratively through a material balance approach or it can be measured directly if a dynamic saturation measurement system (Ayub and Bentsen (2000 and 2001)) is used in a core-flooding experiment. When an iterative scheme is required, it is included or incorporated into the solution procedure in order to be able to estimate the inlet saturation successfully as a function of time. This solution procedure is presented later in Section 5.8. The iterative scheme is similar to the scheme presented by Shen and Ruth (1994a) with a slight modification to automatically re-calculate $S*$ in situations where an unsatisfactory value of $S*$ has been estimated. In the alternative option, for a given set of laboratory experiments, $S*$ data obtained through dynamic saturation measurement can be fitted against τ, using the exponential equation (Equation 5.51), presented by Shen and Ruth (1996) to obtain the fitting coefficients:

$$S^*(\tau) = a_o\left\{1.0 - b_o.\exp\left[-\left(c_o\frac{\tau}{N_c}\right)^{d_o}\right]\right\} \text{........................(5.51)}$$

where $a_o (\geq 1)$, and b_o, c_o and d_o are the fitting coefficients. Equation (5.51) with the fitting coefficients can then be used to obtain any value of $S*$ at any given τ.

5.8. Discretization Scheme and Solution

The implicit finite difference discretization scheme is employed for the computation of fractional flow and saturation profiles. Theoretically, the implicit scheme eliminates the need to establish the stability criteria since the formulation is unconditionally stable in a certain sense. However, this formulation is more mathematically and computationally intensive than the explicit scheme. The main advantage of the scheme presented here is that the Lagrangian formalism eliminates the need for space discretization thereby reducing computation time.

Also, preference for the finite difference method over the finite element method is subjective, because neither of the two can be said to be clearly superior. The finite difference method is more appropriate for handling differential equations that pertain to petroleum engineering problems, especially time-based problems. It is also much easier to implement. It requires less time and less computer storage space. In the finite element-based solution, the computational effort that is required for the matrix coefficients is much more than that required for the finite-difference method (Young (1978)). Accurate solutions have been demonstrated with finite element methods. However, "competitiveness with finite difference has not been established for most nonlinear reservoir simulation problems" (Young (1978)). The other advantage of the finite element method is that it can handle more complicated geometries easily.

In solving the relevant equations above, the aim is to be able to solve for the fractional flow, pressure (or pressure gradient) and saturation as functions of time and space. More specifically defined, the objectives of solving the equations are to be able to predict: 1) fractional flow versus saturation with time as a cross plot, 2) saturation versus space with time as a cross plot, 3) potential (or potential gradient) versus distance. Specifically, solving Equations (5.16), (5.41) and (5.45), using the stated initial and boundary conditions, must be undertaken, if one is to be able to meet these objectives.

5.8.1. Flow Equation

Equation (5.45) is a second order, non-linear parabolic partial differential equation that cannot be solved analytically. The equation can be solved by the finite difference numerical scheme described below. Equation (5.45) may be discretized using an implicit finite difference scheme as follows:

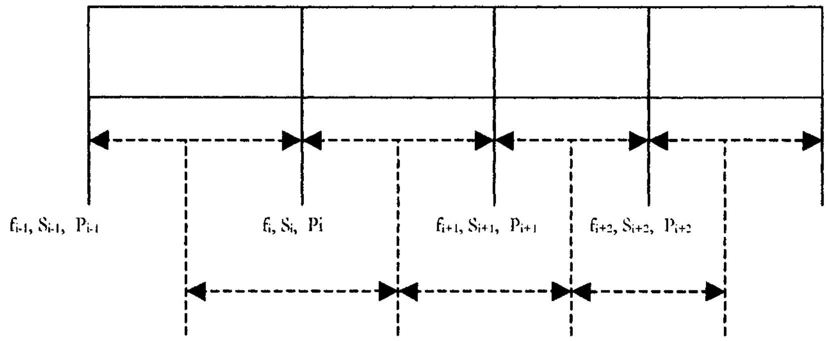
$$f_i^{n+1} - f_i^n = (f_i^{n+1} - G_{i+1/2}^{n+1})^2 (f_{i+1}^{n+1} - 2f_i^{n+1} + f_{i-1}^{n+1}) \chi_{i+1/2}^{n+1} \text{......................(5.52)}$$

where

$$\chi_{i+1/2}^{n+1} = \frac{\Delta\tau}{N_c C_{i+1/2}^{n+1} \Delta S^2} \text{...(5.53)}$$

This form of implicit finite difference approximation scheme is called backward-in-time, center-in-saturation, which is analogous to backward-in-time, center-in-space - BTCS (Hoffmann (2001)). The scheme is unconditionally stable but the time step is restricted only by accuracy requirements, such as deviation error from experimental or analytical results.

The term, $n+1$, denotes the time step after the previous time step, n, and the term f_i^{n+1} is the fractional flow at point, i , at the $n+1$ time step. In Equations (5.52 and 5.53), the fractional flow and the other fluid and reservoir properties are specified at the block ends or at the grid points. This is depicted in Figure 5.1. In Figure 5.1, the subscripts, $i+\frac{1}{2}$, $i+\frac{3}{2}$, $i+\frac{5}{2}$, etc., on the parameters, $C(S)$ and $G(S)$ represent the lumped properties of the i, $i+1$, $i+2$, etc. grid points, respectively. They are measured at the saturation block ends.



f_{i-1}, S_{i-1}, P_{i-1}
f_i, S_i, P_i
f_{i+1}, S_{i+1}, P_{i+1}
f_{i+2}, S_{i+2}, P_{i+2}

Figure 5.1: Schematic representation of domain discretization (saturation)

Rearranging Equation (5.52) gives:

$$D_i(f_{i+1}, f_i, f_{i-1})^{n+1} = (f_i^{n+1} - G_{i+1/2}^{n+1})^2 (f_{i+1}^{n+1} - 2f_i^{n+1} + f_{i-1}^{n+1}) \chi_{i+1/2}^{n+1} - f_i^{n+1} + f_i^n = 0 \text{......(5.54)}$$

In Equation (5.54), $D(f)$ refers to D as a function of three independent variables, f_{i-1}^{n+1}, f_i^{n+1} and f_{i+1}^{n+1}. Equation (5.54) can be written for each saturation point, i, forming a system of non-linear equations that can be solved using the **Newton-Raphson** iterative method described as follows. For convenience sake, henceforth the superscripts $n+1$ on the fractional flow and other variables in Equation (5.54) are dropped and anywhere the variables are encountered, it shall be assumed that they are at time $n+1$ unless otherwise stated. Also, for clarity's sake, let the fractional flow values at time, n, which are known values, be represented by $f*$. That is, $f_{i-1}^n, f_i^n, f_{i+1}^n \Rightarrow f_{i-1}^*, f_i^*, f_{i+1}^*$, respectively. Equation (5.54) then becomes:

$$D_i(f_{i+1}, f_i, f_{i-1}) = (f_i - G_{i+1/2})^2 (f_{i+1} - 2f_i + f_{i-1})\chi_{i+1/2} - f_i + f_i^* = 0 \text{ ....}(5.55)$$

Taking partial derivatives of D_i with respect to f_{i-1}, f_i and f_{i+1} give respectively:

$$\frac{\partial D_i}{\partial f_{i-1}} = (f_i - G_{i+1/2})^2 \chi_{i+1/2} = 0 \text{ ....}(5.56)$$

$$\frac{\partial D_i}{\partial f_i} = 2\chi_{i+1/2}(f_i - G_{i+1/2})[f_{i+1} - 3f_i + f_{i-1} + G_{i+1/2}] - 1 = 0 \text{ ....}(5.57)$$

$$\frac{\partial D_i}{\partial f_{i+1}} = (f_i - G_{i+1/2})^2 \chi_{i+1/2} = 0 \text{ ....}(5.58)$$

Equations (5.56), (5.57) and (5.58) form a system of equations, which can be written as a Jacobian, J, for a system of non-linear equations derived from Equation (5.55) as follows:

$$J = \left(\begin{array}{cccccc} \frac{\partial D_1}{f_i} & \frac{\partial D_1}{f_{i+1}} \text{....} & i=1 & & & \\ & \frac{\partial D_2}{f_{i-1}} & \frac{\partial D_2}{f_i} & \frac{\partial D_2}{f_{i+1}} \text{....} & i=2 & \\ & & \frac{\partial D_3}{f_{i-1}} & \frac{\partial D_3}{f_i} & \frac{\partial D_3}{f_{i+1}} \text{....} & i=3 \\ & & \vdots & \vdots & \vdots & \\ & & \vdots & \vdots & \vdots & \\ & & & \frac{\partial D_n}{f_{i-1}} & \frac{\partial D_n}{f_i} \text{....} & i=n \end{array} \right) \text{ ....}(5.59)$$

Hence, by computing the Jacobian matrix of the system of non-linear equations, the approximate solutions of f_{i-1}, f_i and f_{i+1} $(\Rightarrow f_j,\ j = 1,2,3....n)$, denoted by $\tilde{f}_{i-1}$, $\tilde{f}_i$ and $\tilde{f}_{i+1}$, can be found using:

$$\tilde{f}_{i-1} = f_{oi-1} - f_{i-1} \text{ ....}(5.60)$$

$$\tilde{f}_i = f_{oi} - f_i \text{ ....}(5.61)$$

$$\tilde{f}_{i+1} = f_{oi+1} - f_{i+1} \text{ ....}(5.62)$$

where f_{oi-1}, f_{oi} and f_{oi+1} refer to the initially assumed or previous iteration values of the f_i s and f_{i-1}, f_i and f_{i+1} are computed from a linear system of the form, $Jf = D$ given by:

$$J(f_{oi-1}, f_{oi}, f_{oi+1})^* \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ f_n \end{pmatrix} = \begin{pmatrix} D_1(f_{oi-1}, f_{oi}, f_{oi+1}) \text{....} & i=1 \\ D_2(f_{oi-1}, f_{oi}, f_{oi+1}) \text{....} & i=2 \\ D_3(f_{oi-1}, f_{oi}, f_{oi+1}) \text{....} & i=3 \\ \vdots & \\ \vdots & \\ \vdots & \\ \vdots & \\ D_n(f_{oi-1}, f_{oi}, f_{oi+1}) \text{....} & i=n \end{pmatrix} \text{ ....}(5.63)$$

Complete listings of the Jacobian matrix and the RHS of Equations (5.63) are presented in Appendix A2. The matrix in Equation (5.63) is a tri-diagonal matrix and can be solved using the "Thomas algorithm" technique or any other iterative method like "Gauss-Siedel" or "PSOR - point-wise successive over relaxation" method (Thomas (1982) and Press et al. (1992)). In this work, the **Thomas algorithm** technique is employed. If the solutions of $\tilde{f}_{i-1}$, $\tilde{f}_i$ and $\tilde{f}_{i+1}$ are within acceptable margins, the solution procedure is terminated; otherwise, the values of $\tilde{f}_{i-1}$, $\tilde{f}_i$ and $\tilde{f}_{i+1}$ are substituted back into Equation (5.63) as f_{oi-1}, f_{oi} and f_{oi+1}, respectively, and the equation is solved again. This continues until the differences between successive approximations are small enough or within an acceptable margin.

In the Newton-Raphson method above, the solution of the system of non-linear equations is obtained by an outer iterative method, which linearizes the system, and an inner method, which solves the system of resulting linear equations. An alternative strategy is to obtain the solution of the system of non-linear equations directly iteratively. This technique is similar to the Newton-Raphson method except that the solution of the f's is obtained directly iteratively using the equation below:

$$f_{i+1}, f_i, f_{i-1} = \frac{D_i\left(f_{oi-1}, f_{oi,} f_{oi+1}\right)}{\frac{\partial D_i}{\partial f_i}\left(f_{oi-1}, f_{oi,} f_{oi+1}\right)} \qquad (5.64)$$

This method is the non-linear Jacobi version of the NLOR (non-linear over-relaxation) method presented by Ames (1992) with the relaxation parameter taken as 1. This method is referred to here as the **Newton-Jacobi** method. Both the Newton-Raphson and Newton-Jacobi methods were implemented as optional solvers in the numerical simulator developed in this research as presented in Section 5.12.

5.8.2. Saturation Grid Size and Time Step Selection

For the implicit solution technique, the selection of ΔS can be arbitrary depending on the number of the saturation grid size (uniform size/structured grid). Generally:

$$0 < \Delta S < 1 \qquad (5.65)$$

and ΔS is generally given as:

$$\Delta S = \frac{S}{Number\ of\ grid\ blocks} \qquad (5.66)$$

where S is the saturation discretization domain. The maximum value of the normalized saturation, $S_{\max}$, occurs at the inlet from the beginning and throughout the displacement process.

Theoretically, the selection of $\Delta\tau$ in Equation (5.53) can be arbitrary because of the implicit finite difference scheme employed. But in reality, selecting $\Delta\tau$ such that a single time step is used can result in an incorrect solution due to accuracy requirements (See Section 5.8.1). An iterative procedure was used in arriving at the actual number of time steps used in the numerical simulation results presented in Chapter 7 through comparisons with experimental results. If an explicit finite difference scheme had been used, the stable time requirement stated in the flow chart of Figure 5.2 would have been used (Bentsen (1978)). This stable time requirement makes the computation run longer on a computer.

5.8.3. Displacement Equation

The finite difference equivalent of the first-order hyperbolic Equation (5.16) can also be written. To avoid or reduce numerical dispersion (front smearing), oscillating solutions (unstable solutions near steep gradients) and grid orientation effects (Jensen (1989), Carr and Christie (1983) and Peaceman (1977)) in the discretization of Equation (5.16), a forward-in-saturation and a center-in-time (Crank-Nicholson - Rice (1983)) finite difference approximation scheme is employed.

The resulting approximated equation is unconditionally stable but the time step is limited by accuracy requirements (Hoffmann (2001)). The approximation is given as follows:

$$\frac{\xi_i^{n+1}-\xi_i^n}{\Delta\tau}=\frac{f_{i+1}^n-f_i^n}{2\Delta S}+\frac{f_{i+1}^{n+1}-f_i^{n+1}}{2\Delta S}\quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5.67)$$

Having calculated the values of the f_i^{n+1}s, the distance traveled by any saturation can then be calculated at time step $n+1$ using Equation (5.67). The algebraic form of Equation (5.67) is given as:

$$\xi_i^{n+1}=\xi_i^n+\Delta\tau\frac{f_{i+1}^n-f_i^n}{2\Delta S}+\Delta\tau\frac{f_{i+1}^{n+1}-f_i^{n+1}}{2\Delta S}\quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5.68)$$

The following plots can then be generated after estimating the distances traveled: 1) fractional flow versus saturation with time as a cross plot and 2) saturation versus space with time as a cross plot. In Equation (5.68), the values of ΔS in both the second and the third terms of the RHS are the same. Also, both Equation (5.54), the discretized form of the fractional flow equation, and Equation (5.68), the discretized form of the displacement equation, are solved on the same grid number and time steps subject to accuracy requirements.

5.8.4 Pressure Equation

Finally, the finite difference equivalent of Equation (5.41) is given as:

$$\frac{\pi_{\psi n_{i+1}}-\pi_{\psi n_i}}{\Delta\xi}=\frac{M_r}{N_c}\left[-\frac{1}{M_r}-1-\left(\frac{f_{2i}}{L\lambda_{r\,2i+1\ 2}^0}-\frac{N_cN^{o2}{}_{i+1\ 2}}{M_r}\frac{\Delta S}{\Delta\xi}+\frac{N_g}{LM_r}\left(\frac{1-\alpha_{2i+1\ 2}}{2R_{12}}\right)\right)\right]\quad\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5.69)$$

The algebraic form of Equation (5.69), with all the variables including the dependent and independent variables at the same time step $n+1$, can be written as:

$$\pi_{\psi t_{i+1}}{}^{n+1} = \left[\frac{M_r}{N_c} \left[-\frac{1}{M_r} - 1 - \left(\frac{f_{2i}}{L\lambda^0_{r2i+1/2}} - \frac{N_c N^{o2}{}_{i+1/2}}{M_r} \frac{\Delta S}{\Delta \xi} + \frac{N_g}{LM_r} \left(\frac{1 - \alpha_{2i+1/2}}{2R_{12}} \right) \right) \right] (\Delta \xi) \right]^{n+1} + \pi_{\psi t_i}{}^{n+1} \quad ....(5.70)$$

This equation is then used to estimate the potential (hence the pressure) for each successive saturation block and distance traveled by making use of all the known variables specified in the equation. The plot of potential (or pressure) versus distance can then be generated. Also, once the velocity distribution along the core is known, the potential (pressure) gradient or the normalized potential (pressure) gradient along the core, in the wetting and nonwetting phase, can be computed directly with Equations (5.34) and (5.35), respectively.

5.9 Treatment of the Steady State Solution/Pseudo-Steady Solution

The treatment of the steady state is similar to that of the unsteady state solution except that the capillary pressure gradient now degenerates to zero, hence making the definition of the terms contained in the normalized variables different. This is also applicable to the pseudo-steady state condition of varying time-dependent saturation but a constant saturation profile after breakthrough. Unlike in the case of pseudo-steady state flow, for steady state flow the fractional flow no longer changes with time. Essentially, at this condition, Equation (5.45) becomes:

$$(f - G))^2 \frac{\partial^2 f}{\partial S^2} = 0 \quad ....(5.71)$$

The solution of Equation (5.71) is given as:

$$f = G\,, \qquad S_f \le S \le 1 \quad ....(5.72)$$

or

$$\frac{\partial^2 f}{\partial S^2} = 0\,, \qquad 0 \le S < S_f \quad ....(5.73)$$

Equations (5.72) and (5.73) give the fractional flow as a function of τ and ξ in the range specified. With Equation (5.73), the exact solution, by direct integration twice and by applying the boundary conditions, is given as:

$$f_i = f_f \frac{S_i}{S_f} \quad ....(5.74)$$

where f_f is the fractional flow at the floodfront. Determination of the distance traveled by a given saturation and the pressure (or potential) distribution follows a similar procedure as described previously for unsteady state conditions, although some terms in the previous equations are eliminated. This procedure is quite straightforward and trivial and was not implemented or incorporated into the numerical simulation software.

5.10. Error Analysis

In the solutions given in the previous sections, two types of computational error can be identified. These are the truncation or discretization error and the round off error. The truncation error arises due to approximation of the equation by a finite difference equivalent. This process neglects higher order terms of the Taylor series expansion of the equations leading to discrepancies between the values of the derivatives and the approximation. Using a high number of grids can minimize this error, but there is, however, a practical limit to this because the intensity of the computation increases as the number of grids increases. However, in the computer programming codes, a minimum saturation grid number of 10 is specified which is good enough for a maximum normalized saturation of 1. Using a larger number of time steps can also help to minimize truncation error.

The truncation error in the finite difference approximation of the second order parabolic equation (Equation (5.45)) is of the order (Rice (1983)):

$$O\left(\Delta\tau^2 + \Delta S^2\right) \quad \text{..(5.75)}$$

Similarly, the truncation error in the finite difference approximation (Crank-Nicholson Scheme) of the first order hyperbolic Equation (5.16) is of the order (Rice (1983)):

$$O\left(\Delta\tau^2 + \Delta S^2\right) \quad \text{..(5.76)}$$

The round off error arises due to the computer rounding off part of the number after decimal points. Using higher order decimal places in the computer codes minimizes this error, especially when computations occur between numbers having very low and very high magnitudes.

5.11. Algorithm for Computation

The algorithm for coding, in flow chart and descriptive format, is presented in Figure 5.2. It provides details of how the codes interact within various segments when run on a computer. The figure is self-descriptive and provides a general understanding of the simulation process as applied to the present work. It should be noted that the parametric equations used in representing relative permeability and capillary pressure functions in Figure 5.2 are selected such that, dK_{r1}/dS and $d\pi_c/dS$ behave in such a way that at $S = 0$, $\partial f/\partial S$ is finite; that is, greater than zero (Shen and Ruth (1994a)).

5.12. Programming/Codes Development

The numerical simulation process detailed above is codified into a standalone simulator. The solution techniques described above were codified as optional objects into a standalone interfacial coupling simulator (ICS) executable using the Java^{TM} programming language (Gosling et al. (2000) and Lindholm and Yellin (1999)). Originally, the development of the simulator started with the Visual Basic programming language, but this was later changed to Java due to ease of use and wide availability of Java resources. The GUI of the interfacial-coupling simulator is shown in Figure 5.3. The source codes (.java files) and their byte codes representations (.class files) for the ICS, and the packaged executable JAR (Java Archive) file - ICS.jar, are contained in the CD-ROM at the back of this thesis. The source codes listings are also contained in Appendix A1.

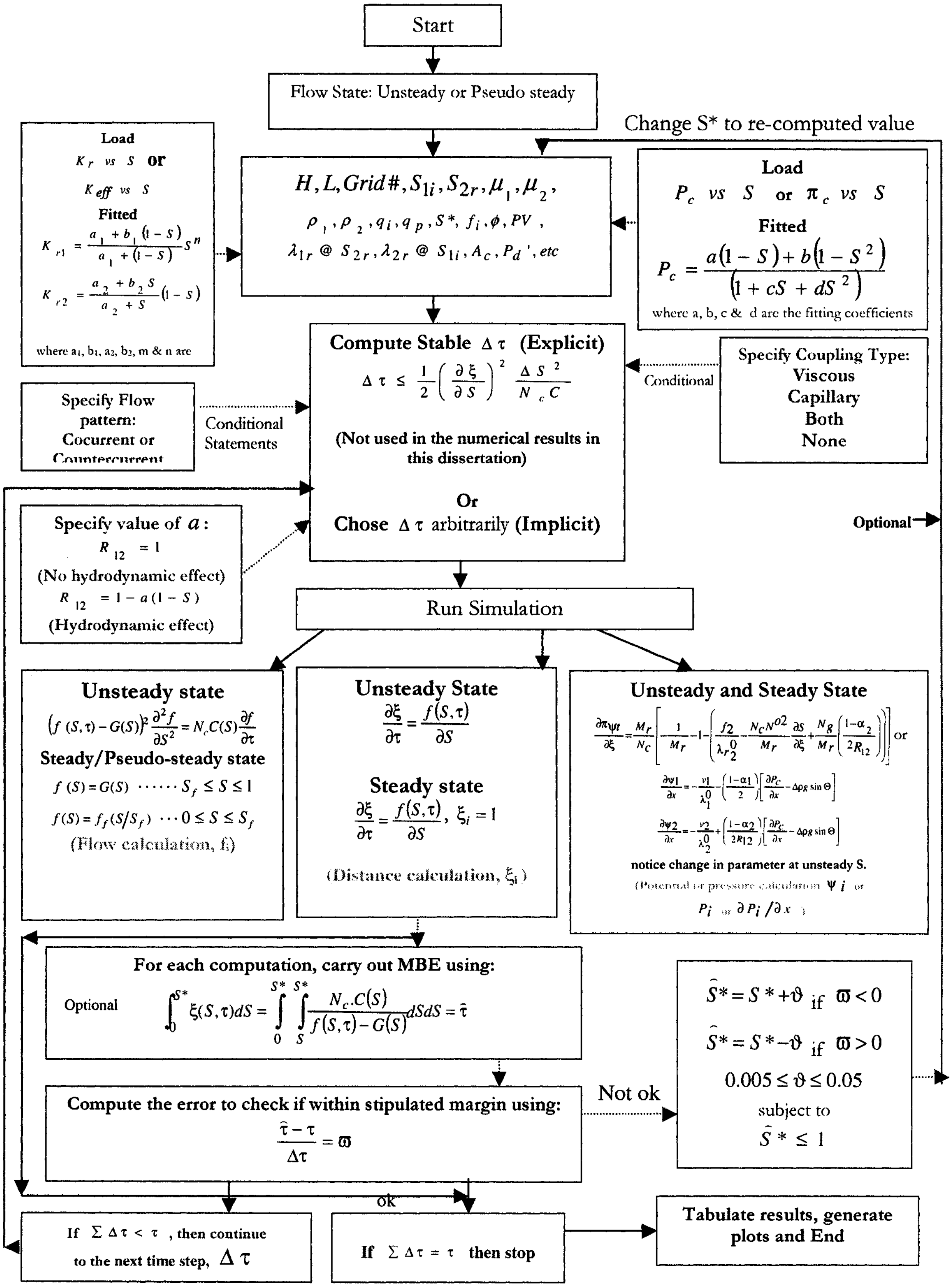
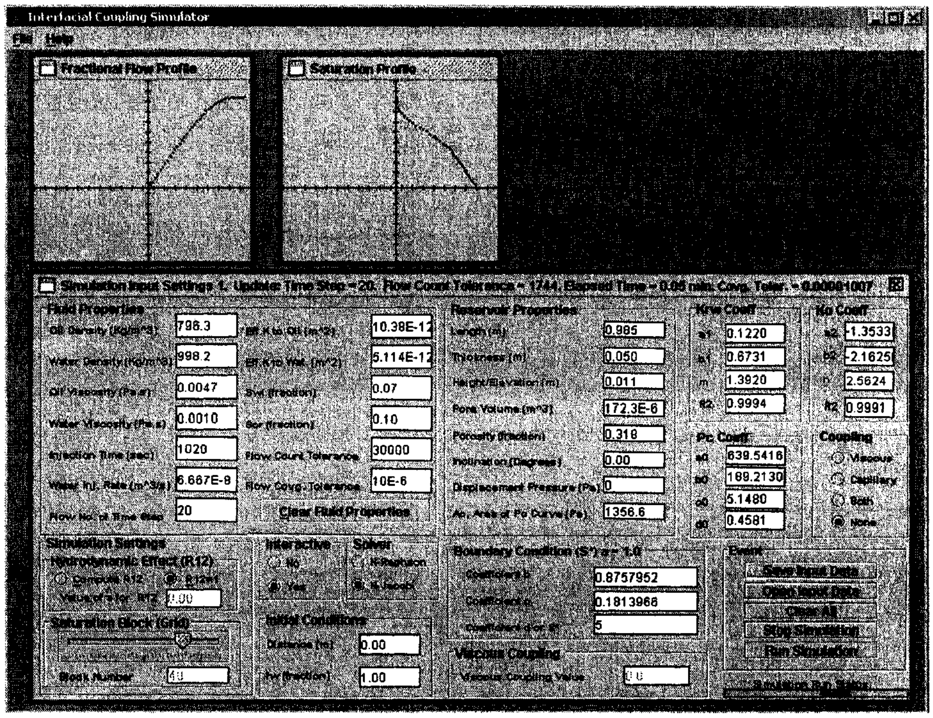

Start
Flow State: Unsteady or Pseudo steady
Change S* to re-computed value
$H, L, Grid\#, S_{1i}, S_{2r}, \mu_1, \mu_2,$
$\rho_1, \rho_2, q_i, q_p, S^*, f_i, \phi, PV,$
λ_{1r} @ S_{2r}, λ_{2r} @ S_{1i}, A_c, P_d ', etc
Load
K_r vs S or
K_{eff} vs S
Fitted
$K_{r1} = \frac{a_1 + b_1(1-S)}{a_1 + (1-S)} S^n$
$K_{r2} = \frac{a_2 + b_2 S}{a_2 + S}(1-S)$
where a₁, b₁, a₂, b₂, m & n are
Load
P_c vs S or π_c vs S
Fitted
$P_c = \frac{a(1-S) + b(1-S^2)}{(1 + cS + dS^2)}$
where a, b, c & d are the fitting coefficients
Compute Stable Δτ (Explicit)
$\Delta\tau \le \frac{1}{2}\left(\frac{\partial\xi}{\partial S}\right)^2 \frac{\Delta S^2}{N_c C}$
(Not used in the numerical results in this dissertation)
Or
Chose Δτ arbitrarily (Implicit)
Conditional
Specify Coupling Type:
Viscous
Capillary
Both
None
Specify Flow pattern:
Cocurrent or
Countercurrent
Conditional Statements
Specify value of a:
$R_{12} = 1$
(No hydrodynamic effect)
$R_{12} = 1 - a(1-S)$
(Hydrodynamic effect)
Optional
Run Simulation
Unsteady state
$(f(S,\tau) - G(S))^2 \frac{\partial^2 f}{\partial S^2} = N_c C(S) \frac{\partial f}{\partial \tau}$
Steady/Pseudo-steady state
$f(S) = G(S) \cdots\cdots S_f \le S \le 1$
$f(S) = f_f(S/S_f) \cdots 0 \le S \le S_f$
(Flow calculation, f)
Unsteady State
$\frac{\partial\xi}{\partial\tau} = \frac{f(S,\tau)}{\partial S}$
Steady state
$\frac{\partial\xi}{\partial\tau} = \frac{f(S,\tau)}{\partial S}, \ \xi_i = 1$
(Distance calculation, ξ_i)
Unsteady and Steady State
$\frac{\partial\pi_{\psi t}}{\partial\xi} = \frac{M_r}{N_c}\left[\frac{1}{M_r} - 1 - \left(\frac{f2}{\lambda_{r2}^0} - \frac{N_c N^{o2}}{M_r}\frac{\partial S}{\partial\xi} + \frac{N_g}{M_r}\left(\frac{1-\alpha_2}{2R_{12}}\right)\right)\right]$ or
$\frac{\partial\psi 1}{\partial x} = -\frac{v1}{\lambda_1^0} - \left(\frac{1-\alpha 1}{2}\right)\left[\frac{\partial P_c}{\partial x} - \Delta\rho g \sin\Theta\right]$
$\frac{\partial\psi 2}{\partial x} = -\frac{v2}{\lambda_2^0} + \left(\frac{1-\alpha 2}{2R12}\right)\left[\frac{\partial P_c}{\partial x} - \Delta\rho g \sin\Theta\right]$
notice change in parameter at unsteady S.
(Potential or pressure calculation ψ_i or P_i or $\partial P_i / \partial x$)
For each computation, carry out MBE using:
Optional
$\int_0^{S^*} \xi(S,\tau) dS = \int_0^{S^*} \int_S^{S^*} \frac{N_c C(S)}{f(S,\tau) - G(S)} dS dS = \hat{\tau}$
Compute the error to check if within stipulated margin using:
$\frac{\hat{\tau} - \tau}{\Delta\tau} = \varpi$
Not ok
$\hat{S}^* = S^* + \vartheta$ if $\varpi < 0$
$\hat{S}^* = S^* - \vartheta$ if $\varpi > 0$
$0.005 \le \vartheta \le 0.05$
subject to
$\hat{S}^* \le 1$
ok
If $\Sigma\Delta\tau < \tau$, then continue to the next time step, $\Delta\tau$
If $\Sigma\Delta\tau = \tau$ then stop
Tabulate results, generate plots and End

Figure 5.2 : Simulation algorithm for interfacial coupling in flow chart format



□ Simulation Input Settings 1. Update: Time Step = 20. Flow Count Tolerance = 1744. Elapsed Time = 0.05 min. Covg. Toler. = 0.00001007

Fluid Properties

Oil Density (Kg/m^3)	798.3	Eff. K to Oil (m^2)	10.38E-12
Water Density (Kg/m^3)	998.2	Eff. K to Wat. (m^2)	5.114E-12
Oil Viscosity (Pa.s)	0.0047	Swi (fraction)	0.07
Water Viscosity (Pa.s)	0.0010	Sor (fraction)	0.10
Injection Time (sec)	1020	Flow Count Tolerance	30000
Water Inj. Rate (m^3/s)	6.667E-8	Flow Covg. Tolerance	10E-6
Flow No. of Time Step	20	Clear Fluid Properties	

Reservoir Properties

Length (m)	0.985
Thickness (m)	0.050
Height/Elevation (m)	0.011
Pore Volume (m^3)	172.3E-6
Porosity (fraction)	0.318
Inclination (Degrees)	0.00
Displacement Pressure (Pa)	0
Ar. Area of Pc Curve (Pa)	1356.6

Krw Coeff

a1	0.1220
b1	0.6731
m	1.3920
R2	0.9994

Ko Coeff

a2	-1.3533
b2	-2.1625
n	2.5624
R2	0.9991

Pc Coeff

a0	639.5416
b0	189.2130
c0	5.1480
d0	0.4581

Coupling: ○ Viscous ○ Capillary ○ Both ● None

Simulation Settings

Hydrodynamic Effect (R12): ○ Compute R12 ● R12=1

Value of a for R12: 0.00

Interactive: ○ No ● Yes

Solver: ○ N-Raphson ● N-Jacobi

Saturation Block (Grid)

Block Number: 40

Initial Conditions

Distance (m): 0.00

Sw (fraction): 1.00

Boundary Condition (S*) a = 1.0

Coefficient b	0.8757952
Coefficient c	0.1813966
Coefficient d or St	5

Viscous Coupling

Viscous Coupling Value: 0.0

Event

Figure 5.3: Graphical user interface (GUI) of the interfacial coupling simulator

5.12.1. Choice of Java

The use of Java for high performance computing has been and continues to be a subject of intensive debate in the computational science and engineering community. Java is both a programming language and a run-time environment (Boisvert et al. (1998)). Its momentum is high and the programming paradigm and advantages, which Java offers, cannot be overlooked easily. These include architectural neutrality (platform independent) and portability, objected-oriented design, simplicity and generality, robustness, security/safety, automated memory management (through garbage collector), multi-threading, distributed-oriented (that is, network-centric) and availability of varieties of libraries, etc. (Flanagan (1997), Boisvert et al. (1998), Bell and Parr (1999), Morelli (2000) and Friedman-Hill (2001)). These advantages are also some of the reasons why the Java programming language was employed in the development of the interfacial-coupling simulator.

Java has been often perceived as being too slow for serious scientific computation and application when compared with other languages like FORTRAN, C, or C++. With the availability and improvement of a just-in-time compiler (JIT) as well as improvement in the Java run-time systems (Boisvert et al. (1998)), this perception is gradually changing and support for Java for use in numeric computing is growing among the numeric computing community. Also, it is now a common consensus among many professionals that the Java language is not actually the problem in this regard; rather, the problem is its implementation. To this end, a lot of professionals and organizations in this community, led by the Java Grande Forum (http://www.javagrande.org) have been working to refine and remove some of the bottlenecks and performance overheads associated with Java (Philippsen et al. (2001), Oliver et al. (2001), Bonachea et al. (2001), Aridor et al. (2001), Boisvert et al. (2001), Getov et al. (2001), Kielmann et al. (2001), Moreira et al. (2001), Thiruvathukal (2002), and most of the references listed in these papers).

5.12.2. Selection and Implementation of Simulator's Solver

This section focuses on the selection and implementation of the solver for the solution of Equation (5.55). With the Newton-Raphson technique, the solution of Equation (5.55) leads to a tri-diagonal system of equations that can be solved using optimized methods of solving a sparse matrix. These include the Thomas algorithm technique (Thomas (1982) and Press et al. (1992)), or iterative techniques like the "Gauss-Siedel" method and the "PSOR - point-wise successive over relaxation" method (Thomas (1982)). In the present case, the Thomas algorithm is well suited since the system of linear equations is tri-diagonal.

If the solution is to be codified using the Java programming language, there is no publicly available Java package that contains classes or methods specifically for solving the tri-diagonal matrix. The only publicly available package is the Java Matrix Package - Jama (Hicklin et al. (1998)) developed by MathWorks, Inc. and the National Institute of Standards and Technology, USA for dense matrices. As a result of this, a Thomas algorithm method (object) was implemented as a solver. The Thomas algorithm was implemented in Java as a Java object that is re-usable and can be plugged into any Java numeric package as a solver. The implemented Thomas algorithm method (called "ThomasAlgorithmSolution") is for solving a system of linear equations of the form, $Ax = B$, where A is a m x n "tri-diagonal" square matrix. The array of elements in A and B (RHS) are supplied as parameters/arguments to the "ThomasAlgorithmSolution" method/object to compute the solution, E. The source codes for the "ThomasAlgorithmSolution" method are shown in Figure 5.4a.

An alternative option for Equation (5.55) is the Newton-Jacobi method solver discussed in Section 5.8.1. The Newton-Jacobi iterative solution technique is quite straightforward. This was also implemented as a Java method (object) and as an optional solver in the simulator. The method takes two arrays (A and B) as parameters/arguments, which represent the previous iterated solution and its derivates, into the "newtonJacobi" method/object, and uses them to compute the solution, E. The source codes for the Newton-Jacobi method are shown in Figure 5.4b.

5.12.3. Evaluation of the Thomas Algorithm Solver

To evaluate how accurate and how fast this implementation is, its relative performance in terms of solution accuracy and execution time was compared with the publicly available pure Java linear algebra matrix package (Jama - Java Matrix Package, (Hicklin et al. (1998)). The Jama version, 5th August 1998, was used in this evaluation. Evaluations were carried out with Java version 1.4.0 (J2SE version 1.4.0), from Sun Micro System Inc. on a celeron 900 MHz processor, with 256 MEG memory and a Windows NT 4.0 operating system. This version of Java also has the JIT compiler that makes Java codes run faster and efficiently.

```
//Thomas Algorithm method
public double [] ThomasAlgorithmSolution (double upperElements [], double diagonalElements [],
                                double lowerElements [], double rhsElements []){

  //For an m x n matrix tri-diagonal, number of rows (m) = number of cols n = (matrix size)
  int matrixSize = diagonalElements.length;

  //Arrays to hold (primary) main and secondary computed coefficients; and solutions, x
  double A [] = new double [matrixSize];                //Primary
  double B [] = new double [matrixSize];                //Primary
  double C [] = new double [matrixSize];                //Primary
  double D [] = new double [matrixSize];                //Primary
  double W [] = new double [matrixSize];                //Secondary
  double G [] = new double [matrixSize];                //Secondary
  double E [] = new double [matrixSize];                //Solution

  for(int i = 0; i < matrixSize; i++){
    //Primary
    if(i == 0)
      A[i] = 0;
    else
      A[i] = lowerElements[i - 1];

      B[i] = diagonalElements[i];

    if(i == (matrixSize - 1) )
      C[i] = 0;
    else
      C[i] = upperElements[i];

      D[i] = rhsElements[i];

    //Secondary
    if(i == 0){
      W[i] = (C[i] / B[i]);
      G[i] = (D[i] / B[i]);
    }
    else if(i > 0){
      W[i] = (C[i] / (B[i] - (A[i] * W[i-1])));
      G[i] = ((D[i] - (A[i]*G[i-1])) / (B[i] - (A[i] * W[i-1])));
    }
  }

   //Solution
   for(int i = 0; i < matrixSize; i++){
     int j = (matrixSize - 1) - i;

     if(i == 0){
       E[j] = G[j];
     }
     else if(i > 0){
       E[j] = (G[j] - (W[j] * E[j+1]));
     }
   }
  return E;
}
```

a. Thomas Algorithm Code Listing

```
//Newton-Jacobi method
public double [] newtonJacobi(double [] A, double [] B){
  int j = A.length;
  double [] E = new double[j];
  for(int i = 0; i < j; i++){
    E[i] = A[i]/B[i];
  }
  return E;
}
```

b. Newton-Jacobi Code Listing

Figure 5.4: The source codes: (a)"ThomasAlgorithmSolution" method and (b)"Newton-Jacobi" method

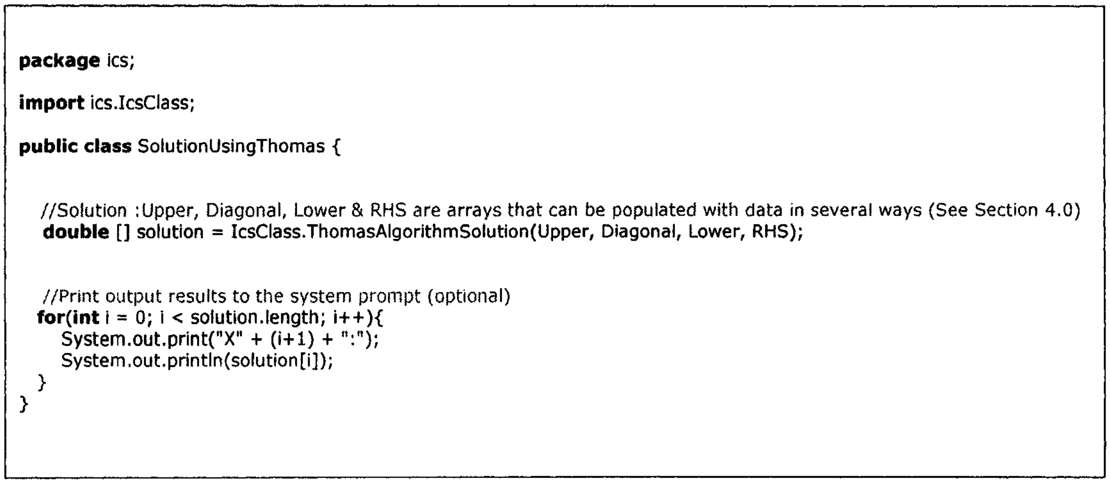
The Java "Matrix" class in the Jama package provides the fundamental operations of numerical linear algebra. "Various constructors in the class create matrices from two-dimensional arrays of double precision floating point numbers" Jama (Hicklin et al. (1998)). Five decomposition classes (Cholesky decomposition of symmetric, positive definite matrices; LU decomposition (Gaussian elimination) of rectangular matrices; QR decomposition of rectangular matrices; Eigenvalue decomposition of both symmetric and non-symmetric square matrices; and Singular value decomposition of rectangular matrices) are implemented. These decompositions are accessed by the "Matrix" class to compute the solutions of simultaneous linear equations and several other matrix functions (Hicklin et al. (1998)).

The snippet codes showing how Jama "Matrix" class can be used as a solver are shown in Figure 5.5. Similar snippet codes for the "ThomasAlgorithmSolution" method are shown in Figure 5.6. Alternative input data sources can be used, for example, from an external file or by direct typing into the source codes or by using the **Math.random** method of the **java.lang** package or by fetching the data from a database.

It is expected that the execution time of the Jama "Matrix" class is going to be much more than that of the "ThomasAlgorithmSolution" method because the Jama package is written specifically for dense matrices and not sparse matrices. But the motivation for the present evaluation is to ascertain the actual magnitude of the differences in execution time and to check if both classes produce the same solution results. Reproducing the same solution results would help to test or ascertain if the "ThomasAlgorithmSolution" method can be used as a solver. The execution time is the clock time required to execute only the computation (excluding matrices' population with data, validation and I/O) using the **System.currentTimeMillis** method available in the **java.lang** package (Bull et al. (2000)). The sizes of the matrices used in the evaluation are selected such that the time of execution is long enough for the **System.currentTimeMillis** resolution to be able to capture it. To compute the execution time, the **System.currentTimeMillis** statement is inserted before and after the solution objects in Figures 5.5 and 5.6 and a variable is declared to consume the difference between the two statements as shown in the snippet codes of Figure 5.7

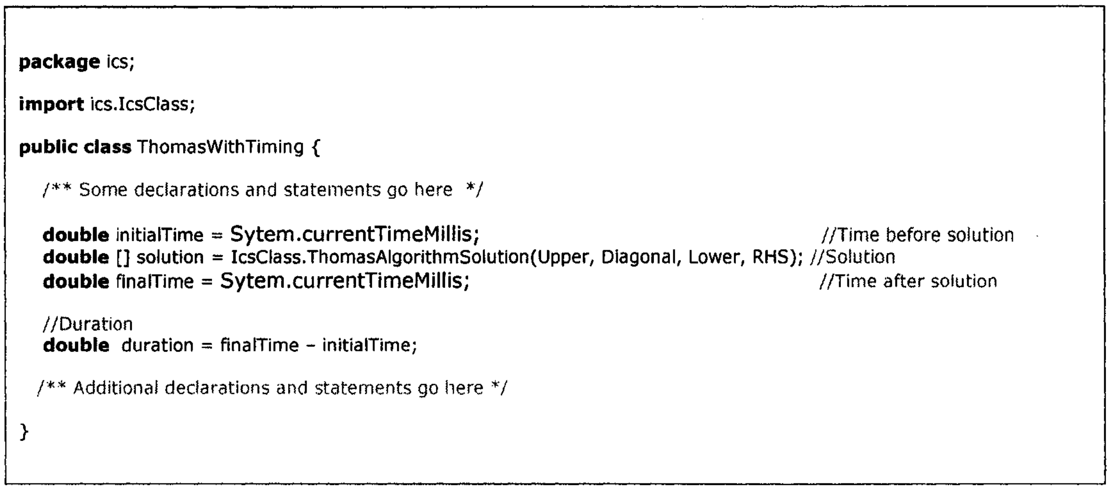
```
package ics;

import Jama.*;

public class SolutionUsingJama {

   //Declare "Matrix" as new objects for the LHS (A) and the RHS (B)
    Matrix A;    Matrix B;

    //Solution = E: A and B are "Matrix" objects can be populated with data in several ways
   Matrix E = A.solve(B);

   //Size of E
   int sizeOfE = E.getRowDimension();

   //Print output results to the system prompt (optional)
   for(int i = 0; i < sizeOfE; i++){
     System.out.print("X" + (i+1) + ":");
     System.out.println(E[i]);
  }
}
```

Figure 5.5: Snippet codes showing the use of the Jama "Matrix" class as a solver



```
package ics;

import ics.IcsClass;

public class SolutionUsingThomas {

   //Solution :Upper, Diagonal, Lower & RHS are arrays that can be populated with data in several ways (See Section 4.0)
   double [] solution = IcsClass.ThomasAlgorithmSolution(Upper, Diagonal, Lower, RHS);


   //Print output results to the system prompt (optional)
  for(int i = 0; i < solution.length; i++){
     System.out.print("X" + (i+1) + ":");
     System.out.println(solution[i]);
  }
}
```

Figure 5.6: Snippet codes showing the use of the "ThomasAlgorithmSolution" method as a solver



```
package ics;

import ics.IcsClass;

public class ThomasWithTiming {

   /** Some declarations and statements go here  */

   double initialTime = Sytem.currentTimeMillis;                                          //Time before solution
   double [] solution = IcsClass.ThomasAlgorithmSolution(Upper, Diagonal, Lower, RHS); //Solution
   double finalTime = Sytem.currentTimeMillis;                                            //Time after solution

   //Duration
   double  duration = finalTime - initialTime;

  /** Additional declarations and statements go here */

}
```

Figure 5.7: Snippet codes showing timing of the "ThomasAlgorithmSolution" method

5.12.4. Results of Evaluation of the Thomas Algorithm Solver

In the first evaluation, the upper, diagonal, lower and right-hand-side (RHS) elements of 4 x 4 matrices, shown in columns 2, 3, 4 and 5 of Tables 5.1 and 5.2 are used as input data, respectively. Results for each are shown in columns 6 and 7 of Tables 5.1 and 5.2, respectively. From the table, output results for the two options (Jama and Thomas) are essentially the same even to 4 decimal places. This demonstrates that the developed object is very effective in solving tri-diagonal matrices. In the case of the Thomas option, the execution time is very small and cannot be properly timed by the relatively small resolution of the **Sytem.currentTimeMillis** method. Hence, the execution time reported for the Thomas option by the computer is 0.0000 in each case as indicated in the last row of Tables 5.1 and 5.2. For The Jama option, the execution time is 20 milli-seconds for each solution.

Table 5.1: Performance Comparison of Accuracy for a 4 x 4 Square Matrix (1st Test)

i	1st Test (Solution, Xi)					
	Upper Elements	Diagonal Elements	Lower Elements	RHS Elements	Thomas Option's Solution (Xi)	Jama Option's Solution (Xi)
1	-	1.0000	2.0000	4.0000	2.0000	2.0000
2	2.0000	1.0000	3.0000	8.0000	1.0000	1.0000
3	1.0000	2.0000	1.0000	10.0000	3.0000	3.0000
4	1.0000	4.0000	-	7.0000	1.0000	1.0000
Time(milli-sec)					0.0000	10.0000

Table 5.2: Performance Comparison of Accuracy for a 4 x 4 Square Matrix (2nd Test)

i	2nd Test (Solution, Xi)					
	Upper Elements	Diagonal Elements	Lower Elements	RHS Elements	Thomas Option's Solution (Xi)	Jama Option's Solution (Xi)
1	-	1.1128	0.7220	3,126,400	1,453,317.0978	1,453,317.0978
2	0.7220	-2.2858	1.5638	0	2,090,233.7030	2,090,233.7030
3	1.5638	-6.1246	1.5808	0	2,384,295.4685	2,384,295.4685
4	1.5808	-3.0141	-	18,166,250	7,261,755.8698	7,261,755.8698
Time(milli-sec)					0.0000	10.0000

In the second evaluation, the Java **random** method of the **java.lang.Math** class was used to generate pseudo-random numbers as input into the tri-diagonal matrices and the time for running the test (solving the matrices) was timed. It should be noted that using the random numbers method is not the best approach to populate the matrices with data as used in the current evaluation. The advantage of this approach is that it eliminates the need to read data from external sources (files, database or intermediate coefficients' computation) and it can still be relatively accepted as the basis of the present evaluation. The tests were carried out on a celeron 900 MHz processor with 256 MEG memory and a Windows NT 4.0 operating system connected to the web.

The results are depicted in Figure 5.8. The figure shows the average execution time of solving the matrices at various matrices' sizes. In both cases, as the number of the matrix's size increases, the execution time increases. The results show that Jama gives a high overhead. In fact a matrix size of more that 2,035 X 2,035 cannot be run using Jama because the computer memory is not large enough to handle the matrix size. The compiler throws a java.lang.OutOfMemoryError exception, when a matrix size bigger than 2035 X 2035 is used. On the other hand, the Thomas object gives a very low overhead. The Thomas object can handle a matrix of up to 702,700 x 702,700 at about 972 milli-seconds. This is a very large size at lower execution times if compared with the case of Jama of 2,035 x 2,035 at 166,270 milli-seconds. Beyond a matrix size of 702,700 x 702,700, the compiler throws a **java.lang.OutOfMemoryError** exception. This shows that the "ThomasAlgorithmSolution" method is much more efficient and runs faster. Below a matrix size of 2,000 X 2,000, the execution time of the Thomas object becomes so small that it cannot be accurately computed by the relatively small resolution of the **Sytem.currentTimeMillis** method. Below this size, the execution time computed by the method is 0.0000.

Hence, the performance of the Thomas algorithm for solving a tri-diagonal matrix was implemented in Java as a Java object. In terms of solution accuracy and execution time, it was found to be efficient. Due to the accuracy and shorter time interval of execution of the "ThomasAlgorithmSolution" method, it was used as the solver for Equation (5.55). When running the simulator, the "ThomasAlgorithmSolution" method is used as the basis of solution as demonstrated in Figure 5.6. The complete source codes for the "interfacial coupling simulator" (ICS) can be found in the CD-ROM at the back of this dissertation and in Appendix A.

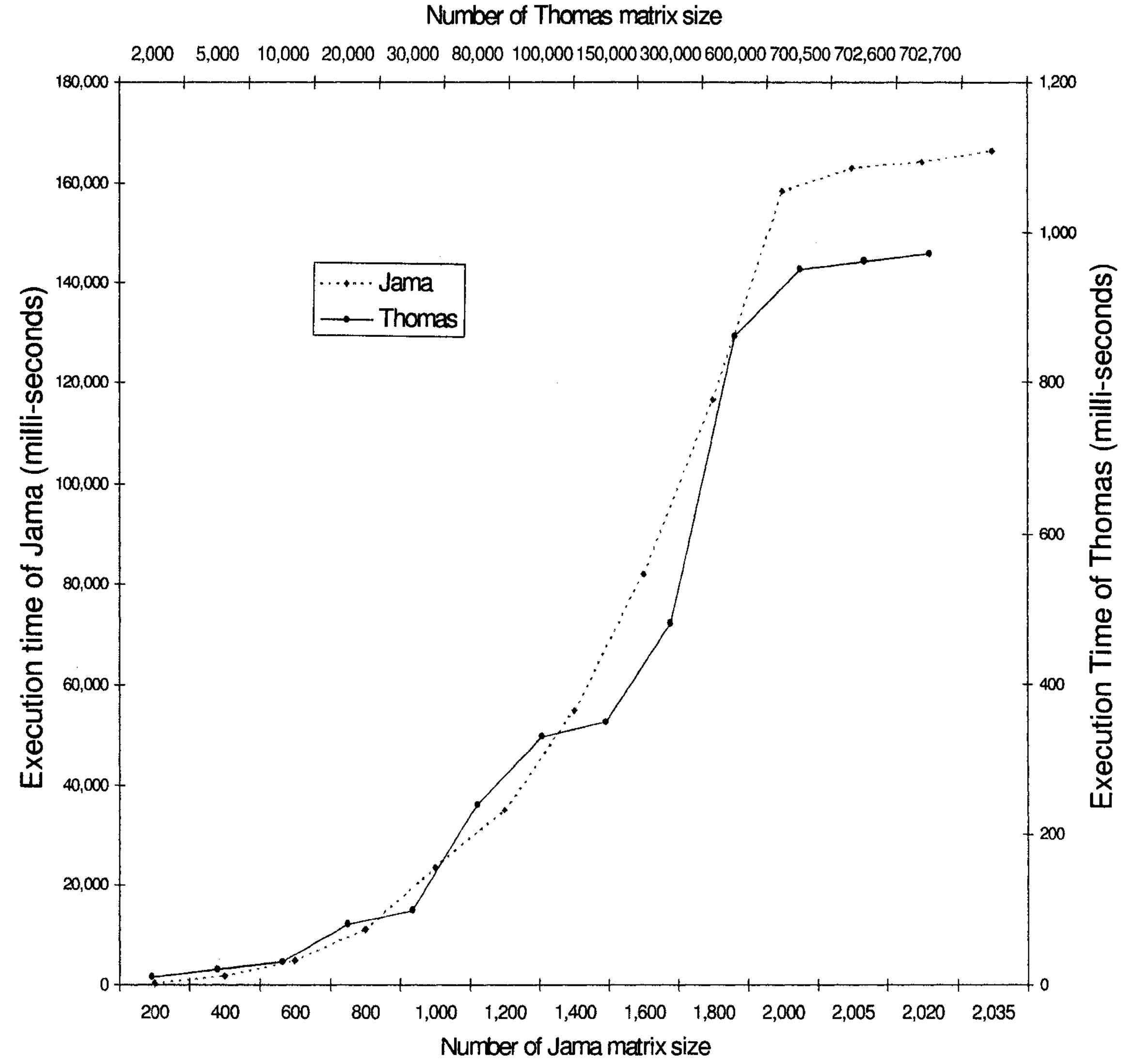


Figure 5.8: Comparing exection time of Jama and Thomas

5.12.5. Description of Source Codes (Classes, Methods and Variables/Fields)

1) **Ics.java**: This is the main class of the software. It is the entrance into the package and it is used in constructing the application. It contains the **static main** method.

2) **IcsFrame.java**: This class contains codes for forming the background GUI that holds the other GUIs. It is the default GUI class that is called by the **Ics.class** at application start up and it also sets the initial GUI state (width, length, location, background color, etc of the main application frame) of the main application window.

3) **IcsClass.java**: This class is the main computational class of the software. The numerical scheme and solution codes are in this class. The class consists of methods, statements and variables for computing various coefficients, intermediate values and final values. Some of the methods have return values while others do not but almost all computation methods are declared as **public** so that they can be instantiated. This class also contains methods and variables for manipulating the GUI components of the software. The main GUI of the software is shown in Figure 5.3. Most of the objects pertaining to the GUI in the source codes are declared **private** to prevent external modifications. In addition, it contains some objects for **i/o** (input-output) operations. These objects include methods for opening and closing input files. These objects (open and close) can be accessed via the GUI components (buttons) in Figure 5.3.

   In addition, the class contains three important specialized methods that are described as follows. These are the "printOutput", "thomasAlgorithmSolution" and the "newtonJacobi" methods. The **printOutput** method is used for handling **output** operations within the computation objects. It contains some variables and associated **exception** handling codes for saving output results within the loop intervals after every computation step. The **thomasAlgorithmSolution** method contains the Thomas algorithm codes. The **newtonJacobi** method contains codes for iterative solution of the system of non-linear equations using the Newton-Jacobi method. Both the **thomasAlgorithmSolution** and the **newtonJacobi** methods are used as the solver for Equation (5.55). The two solvers have a return value (an array object) and other variables for handling solution procedures.

4) **FlowView.java**: This class contains codes for dynamic viewing of the fractional flow and the saturation profiles as the computation progresses. All the methods and variables in the class are declared as **public**. If the "yes" button in the "interactive" panel of the GUI of Figure 5.3 is enabled, the fractional flow and the saturation profiles are plotted and can be viewed dynamically as the computation progresses. Otherwise, if the "no" button is selected, the fractional flow and the saturation profiles cannot be viewed as computation progresses.

5) **IcsFrame_AboutBox.java**: This contains codes for displaying information about author(s), company /university and software version in a dialog box.

Numerical results that were generated using the developed simulator are presented and discussed in Chapter 7.

CHAPTER 6

EXPERIMENTAL EQUIPMENT DESCRIPTION AND MODIFICATIONS

6.1. Introduction

Two sets of laboratory experimental data are used in the analysis presented in Chapter 7. The first set is experimental data reported in the literature (Bourbiaux and Kalaydjian (1990), Manai (1991), Bentsen (1998b and 1997), and Sarma and Bentsen (1989a). The second set is the data acquired in the Instability Laboratory of the School of Mining and Petroleum Engineering at the University of Alberta during the duration of this research.

Full descriptions of the experimental equipment for the first set of data can be found in the literature (Bourbiaux and Kalaydjian (1990), Manai (1991), Bentsen (1998b and 1997), and Sarma and Bentsen (1989a)). This chapter discusses the equipment and modification of the equipment for the second set of data acquired at the University of Alberta.

6.2. Background Information on Equipment

Ayub (2000) developed the equipment for conducting experiments and acquiring the second set of data. Ayub also presented detailed descriptions of the equipment. The picture of the equipment is shown in Figures 6.1. The equipment consists of the core holder (made of acrylic sheet) and injection caps, fluid injection system, effluent collection system, dynamic saturation measurement system, dynamic pressure measurement system and data acquisition systems.

Two types of experiments were conducted. These are USCO (unsteady state cocurrent) experiments - Type I; and SSCO (steady state cocurrent) and USCO (unsteady state cocurrent) experiments - Type II. Due to the poor quality of Type I experimental data, only Type II experimental data were analyzed and presented in Chapter 7. Type I experiment data are shown graphically in Appendix E. Also, the first and second paragraphs of Section 7.2.3.2 discuss the quality issues associated with Type I data. The equipment developed by Ayub, after necessary modification, was used for the acquisition of data from these two types of experiment. A complete description of the equipment is not provided here. But the measurement systems and modifications made to the measurement systems and some other components are presented. A brief discussion of the experimental procedures is also given.

6.3. Measurement Systems

In core flooding experiments, such as those employed in this research, flow rate, pressure and saturation values are measured to describe the displacement process.

6.3.1. Rate Measurement

Measurement of flow rate is done through the use of a mass/weight balance and a timer to determine the interval required in obtaining differences in weight or volume and these data are then recorded and stored in the computer through the data acquisition system.

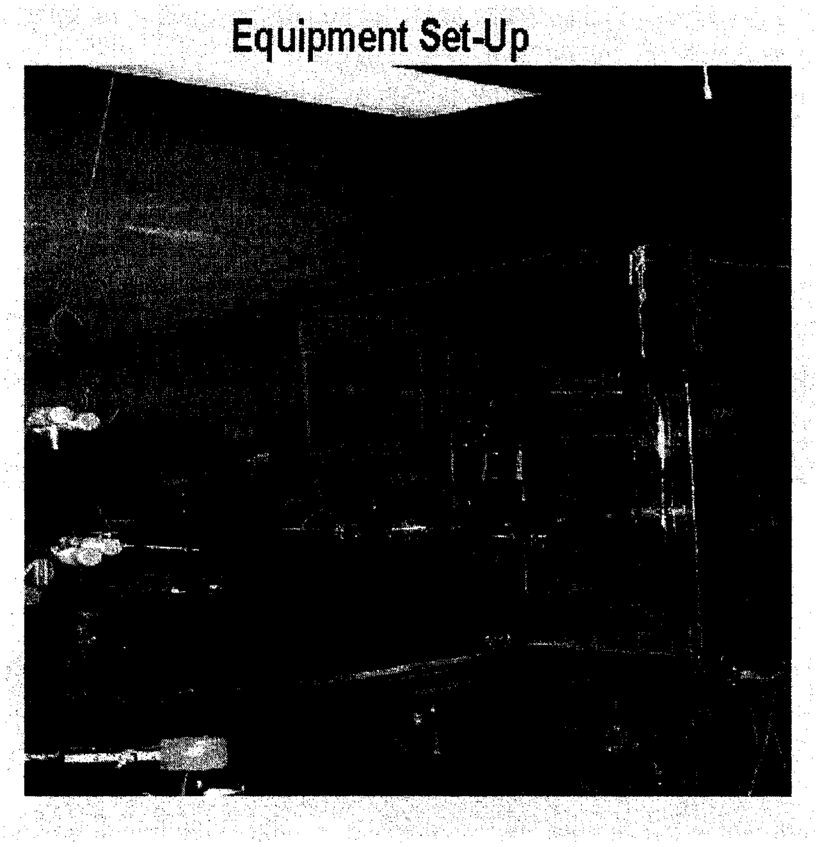


Figure 6.1: Picture of the equipment used for data acquisition

6.3.2. Pressure Measurement

Pressure measurement is done through the use of commercially available sensors or transducers. Such transducers (which are apart of the core flooding equipment) are normally calibrated before a given set of experiments. The pressure in the fluid is measured through the use of porous plates fitted on the intake ports of the transducers. The plates must be wetted and saturated by the fluid, the pressure of which is to be measured through them. Pressure is transmitted to the pressure sensors through these permeable plates.

Pressure differences (and saturation) are measured only for the central portion of the core. In the experiments carried out, differential pressure transducers from Validyne Engineering Corporation are used. Hydrophobic Teflon discs, manufactured by Kontes Glass Company, are fitted at the intake ports of the non-wetting phase transducer to ensure that the transducers sense only non-wetting pressure. On the other hand, the intake ports of the wetting phase transducers are fitted with fritted discs from the Corning Corporation, and membranes (MF disc MCE hydrophilic 0.22μm 47mm) manufactured by Millipore Corporation to make them sense only the wetting phase pressure. The fritted discs provide base supports on which the hydrophobic membranes rest.

6.3.3. Saturation Measurement

Saturation of a core sample can be determined through a variety of methods in the laboratory. Usually, to avoid end effects, only the central part of the core is considered as a tested sample. In certain cases (steady state conditions), saturation values are determined by weighing the sample at every stage of steady state flow; that is, by maintaining a volumetric balance of all the fluids injected and produced from the core sample (Bear (1972)). For unsteady state techniques, a variety of non-invasive techniques for estimating saturation are available. These include X-ray absorption technique (Norel (1964), Geffen and Gladfelter (1952), Laird and Putman (1951 and 1959); the photographic technique (Chatenever and Calhoun (1952), Kimbler and Caudle (1957)); the scanning electronic microscope technique (Swanson (1979); and the fluid solidification and photomicrography technique (Yadav et al. (1987)).

Others include the X-ray computer-assisted tomography (CT) technique (Hove et al. (1985), Vinegar and Wellington (1987), Peters and Hardham (1990), Peters and Afzal (1992), Sprunt et al. (1991), Hicks (1996) and Akin (1999)); the gamma-ray tomography technique (Gilboy et al. (1984), MacCuaig et al. (1986), Vannier and Ellingson (1988) and Ursin (1992)); the nuclear magnetic resonance (NMR) technique (Bates et al. (1983)); the ultrasonic tomography technique (Hiller and Ermert (1984), and Ellingson et al. (1987)); the microwave absorption technique (Sarma (1988) and Manai (1991)) and the technique of measuring changes in electrical properties at several frequency ranges (Bail and Marsden (1957), Sprunt et al. (1991), Berg (1995), Su et al. (2000), Ayub (2000), Adisoemarta (2000), Ayub and Bentsen (2000) and Bona et al. (2001 and 2002)).

Although the X-ray CT tomography technique is the most accurate (Hicks (1996)), the technique of measuring changes in electrical properties is employed in this research. This is because of the high cost of acquiring the X-ray CT scan equipment. Also, the use of the electrical properties' technique is reliable enough within the limits of experimental error. This measurement technique is the same as that employed by Ayub in his previous work with a little modification of the equation used in computing the saturation values from the recorded frequency values and the operational range of the frequency counter (82 to 88 MHz).

In this technique, fluid saturation values are usually determined by inserting electrodes at both ends of a core and measuring the electrical properties of the core (Bear (1972)). The core can also be made to lie between a parallel plate capacitor and the change in electrical properties that results from a change in saturation along the core can then be related analytically to the electrical properties to calculate the actual value of the saturation along the core length. Electrical properties that have been employed in the past include dielectric constant, resistance or conductance, frequency and dissipation constant or factor.

The theory employed by Ayub is based on the capacitance method. The saturation sensing system is a capacitor (resonator) whose dielectric constant or value changes as the wetting phase content in the medium (core holder containing, sand, wetting (water) and nonwetting phases) placed between its parallel plates changes (See Figures 6.2 and 6.3).

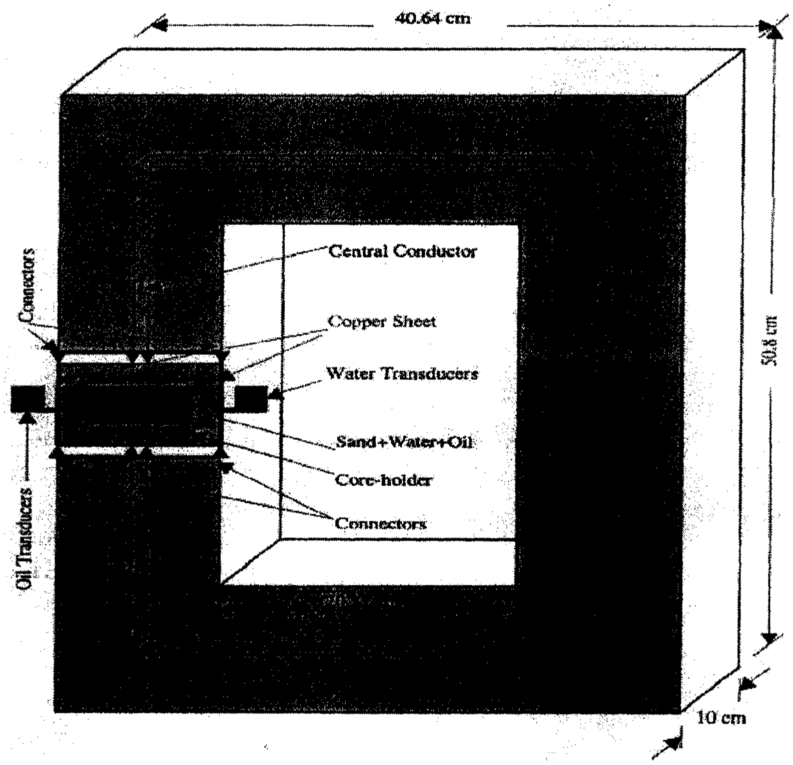


Figure 6.2: Resonator for the saturation measurement system with cross-sectional view of the core holder. Courtesy: Ayub and Bentsen (2000)

The theory on which the resonant measurement is based is for a single component system and it is rather difficult mathematically to determine an equivalent measurement of capacitance of a multi-component system. Core local heterogeneity, electromagnetic interference, temperature and humidity variations in the laboratory might also affect resonator response. Hence, the dielectric constants of the contents of the rectangular core holder (made of acrylic sheet) are lumped together as an equivalent dielectric constant. In the range of frequency employed by Ayub, the change in dielectric constant causes a change in frequency, which is assumed to be due to a change in the wetting phase saturation. The change in dielectric behaviour of such a heterogeneous mixture in the core has a complicated dependence on the frequency and matrix (Arulanandan and Smith (1973)). Ayub used an empirical relationship to determine the saturation profile by simplifying the equation presented by Orlov (1970). This simplified equation is given as:

$$S_1 = \frac{1}{A_o f_o \tan(B_o f_o)} \qquad \text{(6.1)}$$

where

$$A_o = 2\pi Z_o \qquad \text{(6.2)}$$

and

$$B_o = \frac{2\pi l}{c} \qquad \text{(6.3)}$$

In Equations 6.1, 6.2 and 6.3, f_o is the frequency response, $\bar{c}$ is the velocity of light, l is the length of the central conductor of the resonator and Z_o is the characteristic impedance. "The parameter, S_1, is directly equivalent to the capacitance, C, for a specific set of conditions of room temperature and humidity, which are usually assumed as constant" (Ayub (2000)). The parameters A_o and B_o are determined by non-linear regression using SPSS software, once the frequency responses, f_o, against various water saturation levels are determined experimentally. The theory of saturation employed in this work is similar to that presented by Ayub and it is given in the next section with the exception of the polynomial function for curve calibration and the operational range of the frequency.

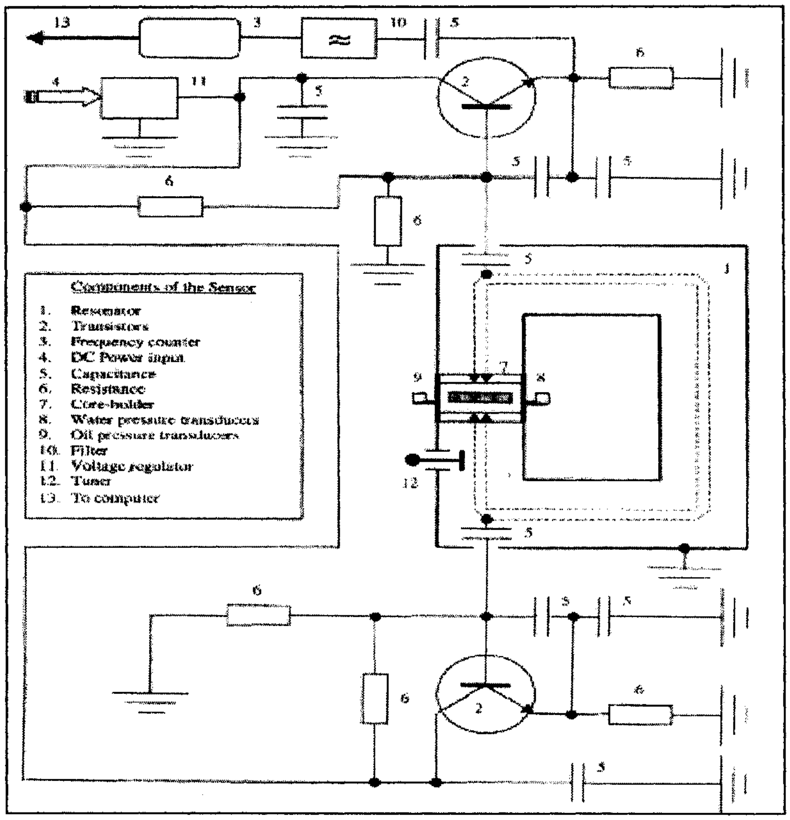


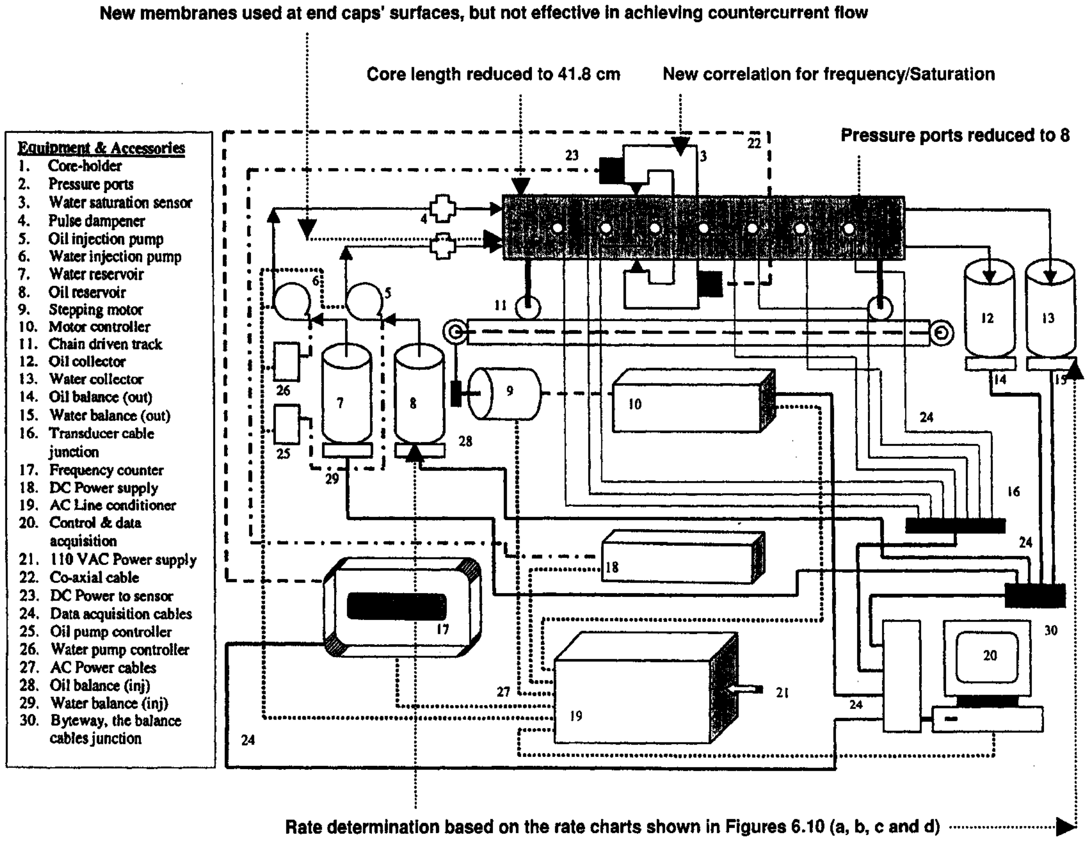
Figure 6.3: Circuit diagram of the water saturation measurement sensor
Courtesy: Ayub and Bentsen (2001)

6.4. Modifications of Components and Calibrations

The necessary modifications to be made on the equipment and experimental set-up and the sensor calibration procedures are given as follows. These modifications are highlighted in Figure 6.4.

6.4.1. Length of the Core

For easier handling, a new core, 41.8 cm long, was used in place of the previous, 100.0 cm long core. The lengths of the center portion of the core, where pressure and saturation measurements were taken after neglecting the end effects region, are 26.5 cm and 18.0 cm respectively. Shortening the length of the core reduces the data density (saturation and pressure) along the core length. This might cause a little, but accepted, reduction in accuracy of the measurements. For analysis purposes, pressure and saturation measurements from 0 to 18 cm of the measurement length were used, because the saturation measurements cover only this interval.



New membranes used at end caps' surfaces, but not effective in achieving countercurrent flow

Core length reduced to 41.8 cm

New correlation for frequency/Saturation

Pressure ports reduced to 8

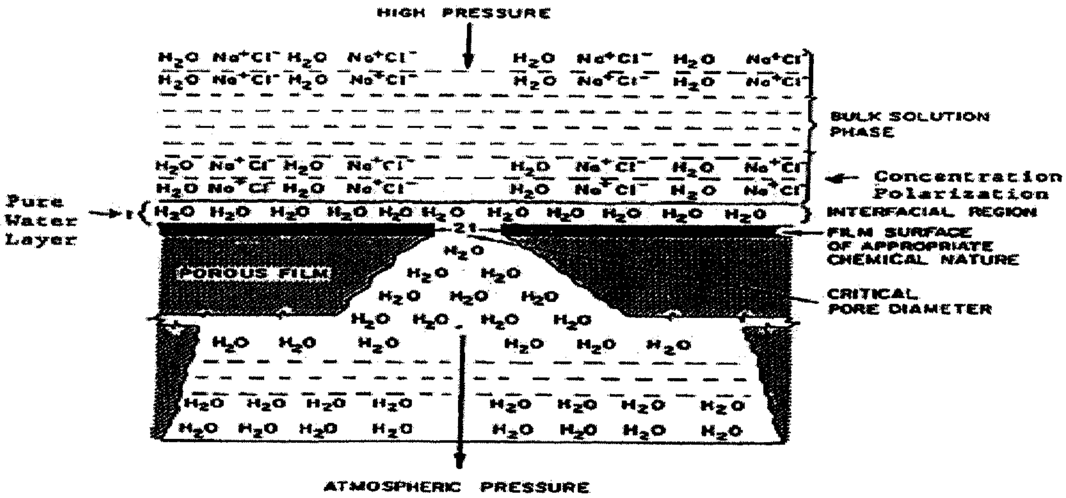
Rate determination based on the rate charts shown in Figures 6.10 (a, b, c and d)

Figure 6.4: Schematic representation of equipment highlighting necessary modifications

6.4.2. Membranes

In the earlier experiments conducted by Ayub, countercurrent flow situations could not be achieved. This was attributed to the change in wettability of the surface membranes at both ends of the core. The membranes were suppose to prevent backward flow of the phases to where they are injected from either side of the core so that countercurrent flow situations can be achieved. It is believed that long exposure to the mixture of wetting and nonwetting phases caused this change in wettability. There was no salt (ions) in the wetting phase (distilled water). Hence, no ionic bond was formed to aid the movement of wetting phase towards the end where it should be produced and prevent it from flowing towards the end where only the nonwetting phase is required to flow through, as shown in Figure 6.5.

As result of this, in this research, tap water, which contains ions, was used as an aid to achieve a perfect bond as shown in Figure 6.5. The introduction of tap water does not introduce any significant error (less than 0.3%) into the saturation measurement, especially if the sensor is calibrated based on the expected frequency response or shift due to tap water and the non-wetting phase. Also, new sets of membranes (MF disc MCE hydrophilic 0.22μm 47mm and Nylon membrane disc hydrophobic 0.22μm 47mm) manufactured by Millipore Corporation were used. Despite this, the membranes did not behave as expected. That is, the membranes did not prevent backward flow of the phases. The membranes could not form a perfect ionic bond that could enhance countercurrent flow of the phases in the opposite direction. Consequently, countercurrent flow experiments could not be conducted.



The figure depicts a "schematic of the preferential sorption-capillary flow mechanism for reverse osmosis separation of sodium chloride from aqueous solutions.

Figure 6.5: Schematic representation of membranes' surfaces chemistry
Courtesy: Osmonics Inc. online products catalog at http://www.osmonics.com/products/Page772.htm

6.4.3. Saturation - Frequency

6.4.3.1. Theory

No final consensus exists on the relation between saturation of the wetting phase (water) and frequency. The relationship between these parameters depends on many factors and several results have been reported at various frequency levels (Davis (1980), Kraszewski (1986), Berg (1995), Nguyen et al. (1999), Su et al. (2000), Adisoemarta (2000), Bona et al. (2001 and 2002) and West et al. (2003)). Depending on the frequency range, from microwave frequency to VHF (very high frequency) of the electromagnetic spectrum, the relationship between a heterogeneous mixture such as fluids/porous media physical properties (porosity, saturation, etc) and electrical properties (dielectric constant, frequency, capacitance, etc) is very complicated.

Davis (1980) concluded from experimental results that oil/water saturation in laboratory porous media can be non-invasively determined by the VHF electrical measurements above 100 MHz. Davis (1980) and Kraszewski (1986) also observed that the anomalous response of fluids in a porous media system at about 100 MHz to 200 MHz was found to be minimal. As a result of this, Ayub conducted dynamic saturation-pressure measurements in this range of frequency. Some electrical components of the resonator were changed. As a result of this, the operating frequency of the frequency counter changed to about 82 to 88 MHz. The frequency measurements taken on the core for 0 to 100 % Saturation ranges from about 85.2181 to 84.8487 MHz. Although Davis recommended operating in the 100 to 200 MHz range, operating in the 82 to 88 MHz can also be accepted. Experimental results presented by Nguyen et al. (1999) from core flooding experiments show that response of frequency to various saturation levels can be observed at 82 to 88 MHz frequencies. Hence the wetting phase saturation can be determined effectively in this range.

To obtain a calibration curve for saturation-frequency to be able to calculate unsteady state saturation values, steady state experimental data were used, covering saturations from the initial water saturation to the residual wetting phase saturation. The average frequency value of 15 frequency data points along the core, at each of the various geometrical and environmentally corrected saturation levels (8 different saturation levels) of the wetting phase, were computed from recorded data. The steady water saturation values were first determined through a volumetric material balance approach. The saturation values were then corrected for geometrical and environmental errors (See Section 6.4.3.2) through the use of Equation (6.1).

The corrected saturation values were then fitted against average frequency, as an inverse third-order polynomial function, with a correlation coefficient of 0.9706. The data used in the calibration curve of Figure 6.6 were obtained when the Type - II (SSCO) experiments were conducted. The calibration curve was used to obtain saturation values for unsteady state experimental data. The relationship between saturation and frequency shown on the curve is:

$$S = 23872.9823 - \frac{2015966.8168}{fq} - \frac{177175448.8055}{fq^2} + \frac{14964400930.2684}{fq^3} \text{......................}(6.4)$$

It should noted that that although Type I and Type II USCO experiments were conducted on different packed cores, the calibration data obtained from Type I SSCO can be used for both because the sand-fluid system for both is the same. Also, their frequency operating range falls in the same interval.

6.4.3.2. Correction for Core Geometry and Environmental Effects

From the previous experiments by Ayub, due to a geometrical problem (non-horizontalness of the core) associated with the construction of the equipment and core holder, the frequency response at each data point along the core deviates slightly from the actual response. Local core heterogeneity, electromagnetic interference, temperature and humidity variations in the laboratory might also contribute to the error in the response. This error also occurred during steady state runs when the response should be the same along the core length, although the profiles for all the saturation values are almost parallel to one another.

The same problem was observed in the present experimental studies as shown in Figure 6.7 for Type II - SSCO runs. A close observation of the steady state saturation profiles as shown in Figure 6.7 shows that the profiles for all the saturation values are also almost parallel to one another, which means that the error is almost the same for each profile. Hence correction can be applied to each profile by lumping the effects at all data points into coefficients A_o and B_o in Equation (6.1).

The coefficients A_o and B_o were determined independently for each saturation profile using the raw frequency measurements and volumetrically-determined steady state saturation with the regression analysis option available in the SigmaPlot Software from SPSS Inc. These coefficients and Equation (6.1) were then used to determine the corrected saturation for each profile. The values of the coefficients A_o and B_o for each profile are shown in Appendix B (Table B1). Raw frequency measurements along the core and the corrected steady state saturation profiles generated with Equation (6.1) are shown in Figures 6.7 and 6.8, respectively. It should be noted that the same problem of deviation of a measured electrical property (resistance) was observed in a previous work by Sprunt et al. (1991) and the shape of the deviation from the correct saturation value is similar to that observed in this research.

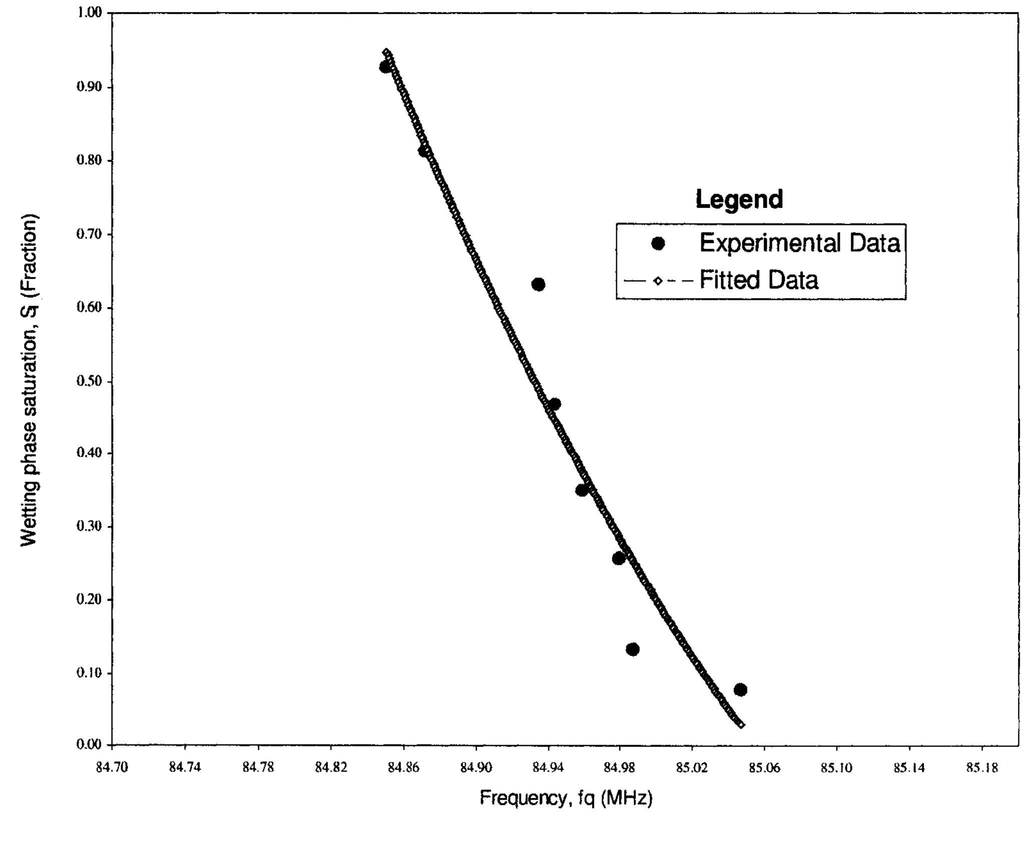


Figure 6.6 : Polynomial saturation calibration curve (Type II - SSCO)

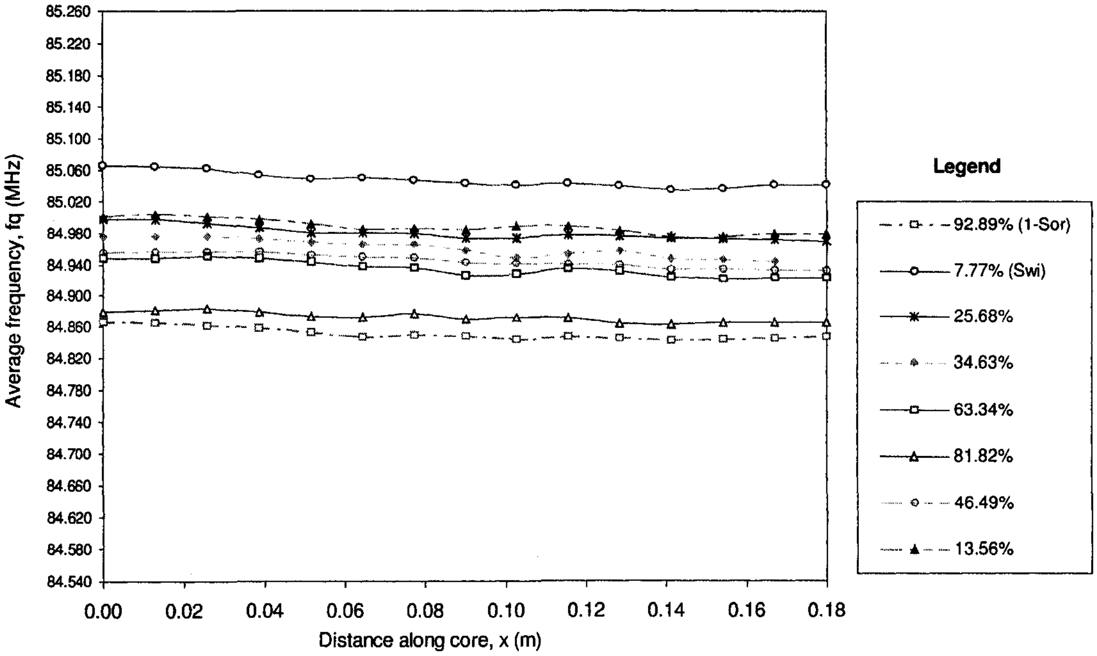


Figure 6.7: Raw frequency measurements profiles along the core at steady state saturation levels (Type II - SSCO)

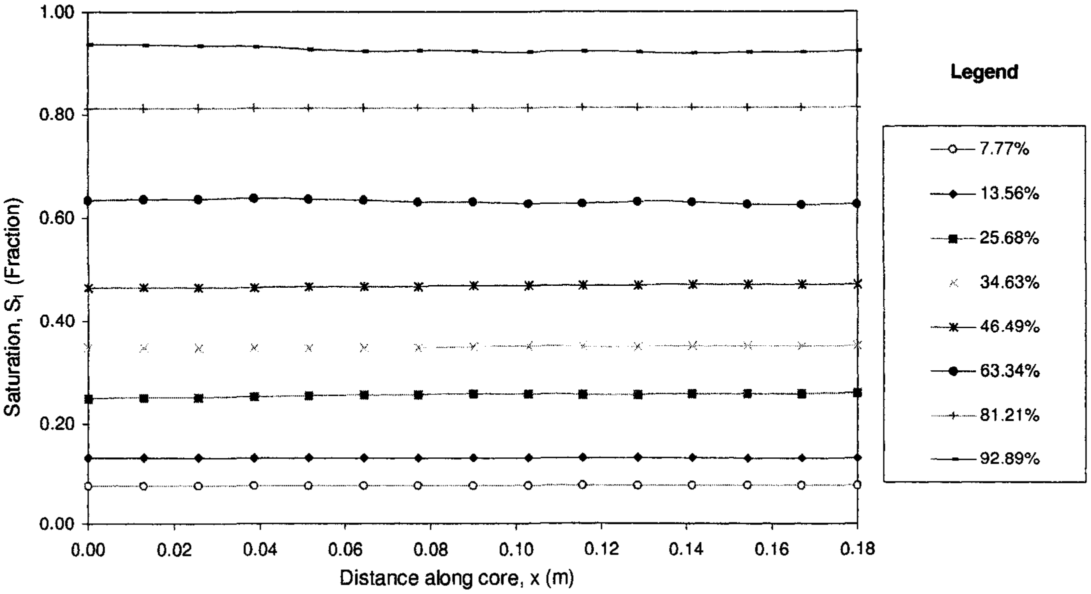


Figure 6.8: Computed steady state saturation profiles along core holder after geometrical and environmental corrections (Type II - SSCO)

6.4.4. Pressure Traducer Calibration

The number of pressure transducers was reduced to 8 because the length of the core was reduced although measurements from only 6 of the pressure sensors, three on either side, were used in the analysis. The Validyne Engineering Corporation is the manufacturer of the pressure transducer. Four pressure transducers are located on either side of the core and spaced as shown in Figure 6.9. Before the transducers were used for pressure measurements they were calibrated using a digital calibrator as opposed to the previous technique of using a cumbersome manual calibration approach.

Out of the total number of 14 pressure transducers used in the previous studies by Ayub, transducer numbers 3, 4, 5 and 7 for the wetting phase, and numbers 9, 10, 12 and 15 for the non-wetting phase, were selected and calibrated. The calibration involves the determination of correlation parameters, namely the intercept, c, and the slope of a straight line, m, that relate the sensor measurements to the calibrator measurements.

The values were then entered into the appropriate input section (pressure transducer calibration section) on the software system. The calibration data and the correlation parameters for the pressure transducers are shown in Appendix C (Table C1). The same calibration data were used for Type I and Type II experiments.

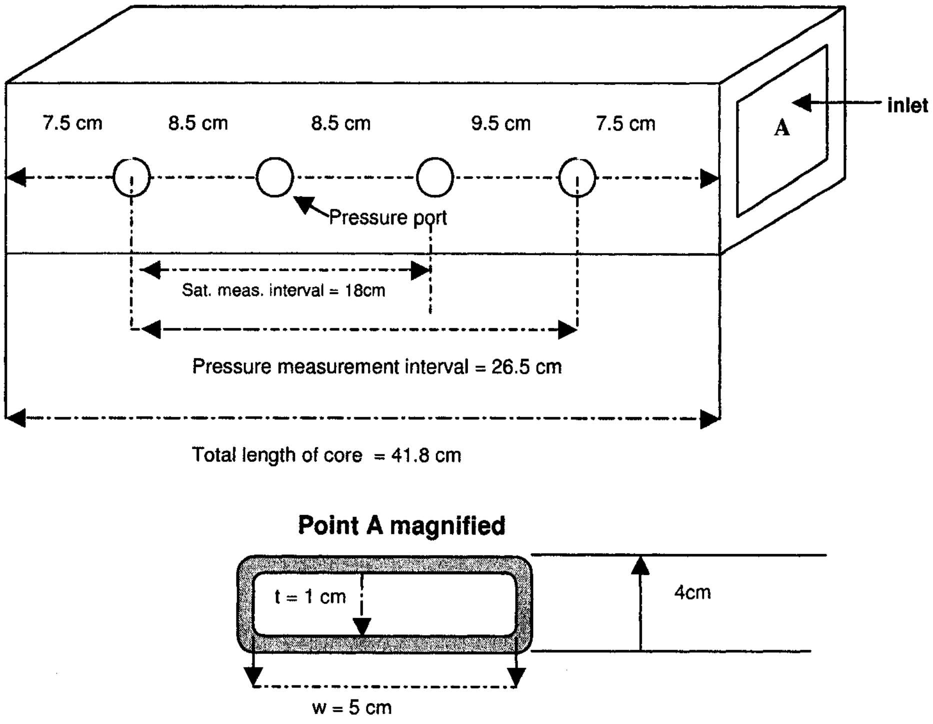

7.5 cm
8.5 cm
8.5 cm
9.5 cm
7.5 cm
Pressure port
A
inlet
Sat. meas. interval = 18cm
Pressure measurement interval = 26.5 cm
Total length of core = 41.8 cm
Point A magnified
t = 1 cm
4cm
w = 5 cm

Figure 6.9: Location of pressure transducers on the core (not to scale).

6.4.5. Flow Rate Determination Chart

The selection of the flow rate of both the wetting phase (water) and non-wetting phase (light mineral oil) greatly affects the saturation profile for steady state experiments. Usually, achieving the desired equally spaced saturation level for a steady state experiment is through a trial and error approach. To ease this cumbersome process, it was decided to calibrate the flow rate of the two pumps (FMI^{TM} pump) used in pumping the phases. This greatly helped in selecting the right combination of stroke length and stroke rate to achieve a desired saturation level.

The calibration was carried out at different stroke rates and stroke lengths for each phase. The pumps were switched on alternatively and the time interval used in pumping a given phase volume is recorded along with the volume at specified pumps' stroke rates and lengths. The values are then plotted as a chart to establish the relationship between flow rate, stroke rate and stroke length. The collected data are plotted as charts in Figures 6.10a, 6.10b, 6.10c and 6.10d. These charts were used as guides when selecting a rate for each of the pumps/phases.

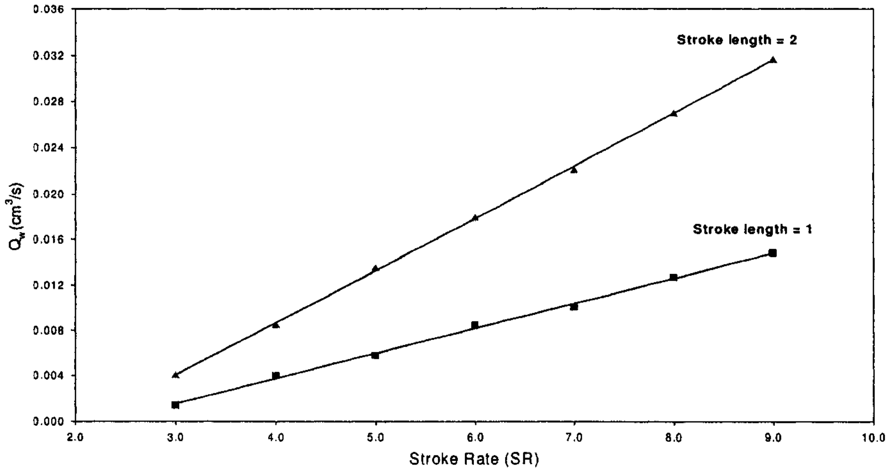


Figure 6.10a: Pump calibration chart for the wetting phase at lower values

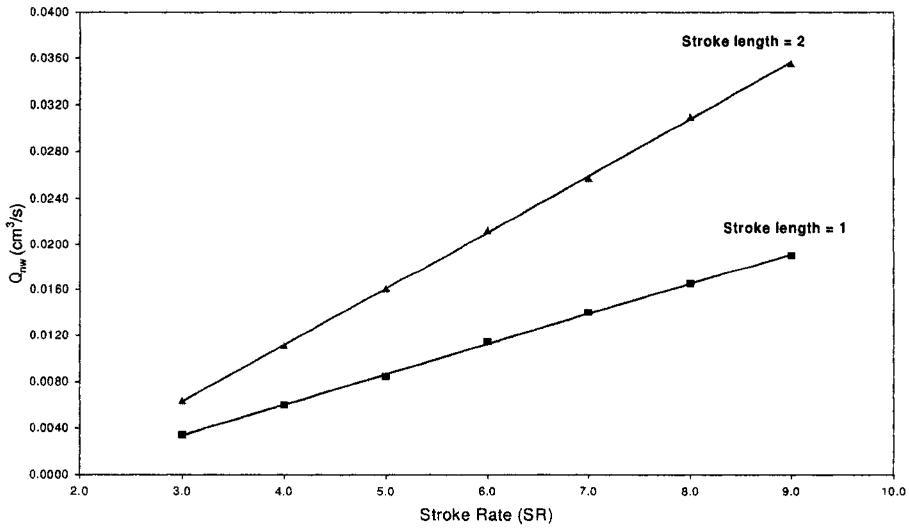


Figure 6.10b: Pump calibration chart for the non-wetting phase at lower values

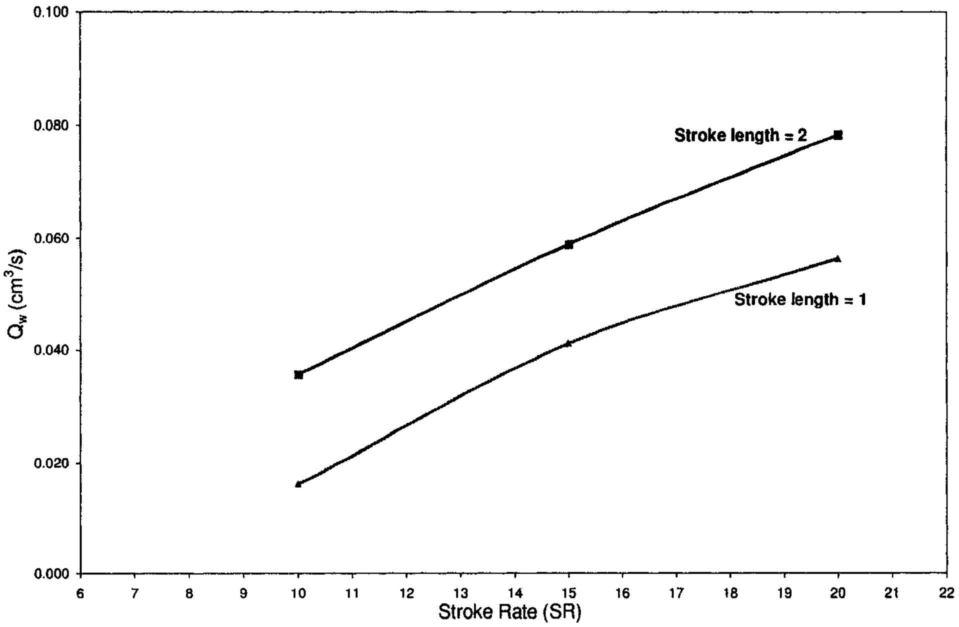


Figure 6.10c: Pump calibration chart for wetting phase at higher values

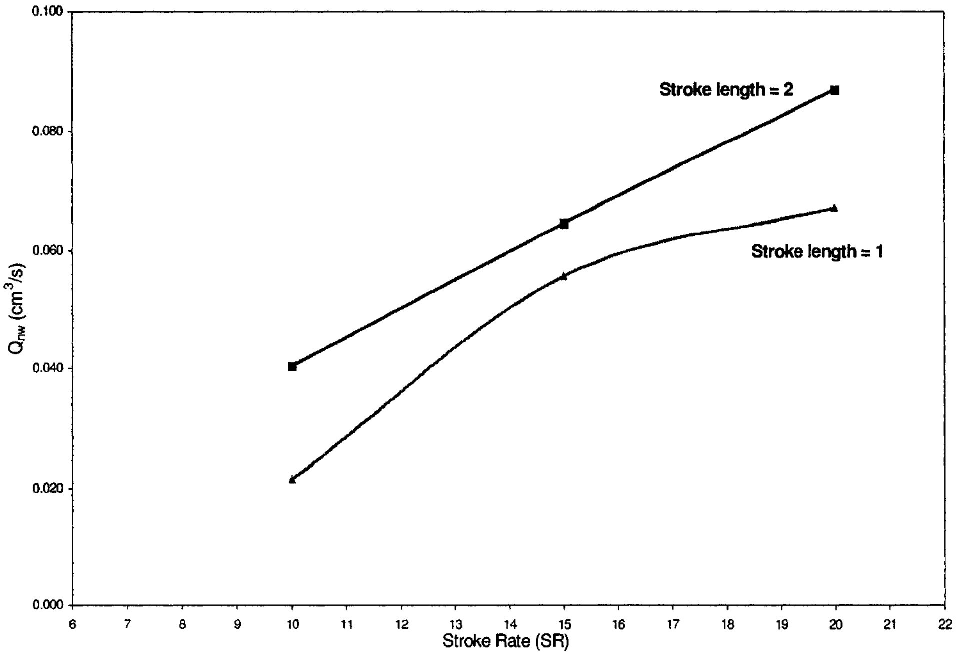


Figure 6.10 d: Pump calibration chart for non-wetting phase at higher values

6.5. Experimental Type and Procedures

6.5.1. Experimental Type

Due to a limitation in the design of the equipment, especially with regards to the use of the right membrane type, as stated in Section 6.4.2, a countercurrent flow experiment could not be achieved. As a result of this, it was decided to acquire only steady state cocurrent (SSCO) and unsteady steady state cocurrent (USCO) data with the equipment after making the necessary modifications discussed in Section 6.4. Data acquired during the experiments are presented and discussed in Chapter 7.

On the other hand, countercurrent flow experimental data used in the analysis presented in Chapter 7 are those obtained from data presented in the literature (Bourbiaux and Kalaydjian (1990), Bentsen (1997 and 1998b), Manai (1991), and Sarma and Bentsen (1989a). Most of these data have some limitations, but they can still be used to verify the modified set of equations developed in Chapter 4.

6.5.1. Summary of Experimental Procedures

Details of the experimental procedures can be found in Ayub (2000, pages 93 - 97). For brevity, these are summarized here. Even though some modifications have been made to the equipment, the procedures followed in acquiring the data are similar to those of Ayub with the exceptions noted in Section 6.4.

Generally, a wet packing procedure is used in all cases. Also, the sand used in packing the core in all cases is Ottawa silica sand of 80-120 mesh size from the Ottawa Industrial Sand Company, USA. The core is allowed to rest and dried through air circulation, after wet packing, for about 1 to 2 days. The drying and weighing procedure is used to establish the porosity of the packed core. The effective permeability to the non-wetting phase at irreducible (connate) wetting phase saturation is first determined using Darcy's equation and the measured rate and pressure data, and fluid properties. This is followed by the unsteady state experiments during which phase pressure, saturation and rate data are acquired as the wetting phase displaces the non-wetting phase.

After the unsteady state experiment, the core is subjected to various constant pre-determined injection rates of both wetting and non-wetting phases simultaneously to establish steady state flow at various saturation levels ranging from irreducible wetting saturation to residual non-wetting saturation. After conducting the last steady state experiment, the core is injected continuously with the wetting phase until the residual non-wetting saturation is achieved. At this point, the effective permeability to the wetting phase at the residual non-wetting saturation is determined using Darcy's equation and the measured data and fluids properties.

Data are acquired through the automated computer acquisition system connected to the intermediate converters and sensors linked to the core holder. The computer acquisition system is made up of the software developed with LabVIEW^{TM}, an intuitive graphical developing tool, and various hardware components, which include interface cards (UPC-L, RS-232, GPIB and DAS-8) installed on the motherboard of the desktop computer. Data acquisition for complete sets of SSCO and USCO experiments on the same core takes approximately 8 days to complete excluding the initial preparation of the core, pressure calibrations and other initial activities. On the other hand, data acquisition for a single set of USCO experiments takes about 6 to 8 hours excluding the initial preparation stages, depending on the desired quantity of data.

CHAPTER 7

MODEL VALIDATION: DISCUSSIONS OF ANALYTICAL, EXPERIMENTAL AND NUMERICAL RESULTS

7.1. Introduction

Three types of experimental data are needed for verification of the modified transport equations in Chapter 4. These are SSCO (steady state cocurrent), USCO (unsteady state cocurrent) and SSCC (steady state counter current) experimental data. Pressure, saturation and rate data were collected as stated in Chapter 6 for SSCO and USCO experiments on the same sand-fluid systems. SSCO, USCO and SSCC data have been collected in the past (Bourbiaux and Kalaydjian (1990), Manai (1991), Bentsen (1998b and 1997) and Sarma and Bentsen (1989a).

Data from the above-cited literature, although having some limitations, can be used to demonstrate the impact of interfacial coupling effects on porous media multiphase flow; that is, they can be used to verify the modified set of transport equations. Hence, both the data collected as reported in Chapter 6 and that reported in the above-cited literature were used in the analysis presented in this chapter. The analysis of how the collected data were used to verify the modified set of equations can be classified into analytical and numerical simulation techniques and these two approaches are described below.

7.2 Analytical Techniques

The modified set of one-dimensional transport equations, which incorporate viscous coupling, capillary coupling and hydrodynamic effects, are given as (from Equations (4.36) and (4.37), allowing for the angle of inclination, Θ, of the core or reservoir):

$$v_1 = -\lambda_1^0\left(\frac{\partial\psi_1}{\partial x} + \left(\frac{1-\alpha_1}{2}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g.\sin\Theta\right]\right) \text{..............(7.1a)}$$

and

$$v_2 = -\lambda_2^0\left(\frac{\partial\psi_2}{\partial x} - \left(\frac{1-\alpha_2}{2R_{12}}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g.\sin\Theta\right]\right) \text{............(7.1b)}$$

The meaning of all the terms in Equations (7.1a) and (7.1b) are given in the nomenclature. These two equations can be verified analytically using two different approaches as follows.

7.2.1. Analytical Approach I

In the first approach, the interfacial coupling coefficients in Equations (7.1a) and (7.1b) can be determined analytically using Equation (4.31), defined as:

$$\lambda_i^* = \alpha_i\lambda_i^0, \quad \text{i} = 1,2 \text{...(7.2)}$$

The variables on the LHS and RHS of Equation (7.2) are measurable directly or derivable from experimental data. Hence, when the values of the variables are substituted into Equation (7.2), the RHS of Equation (7.2) must be approximately equal to the LHS of the equation. It should be noted that the coupling coefficients, α_1 and α_2, in the RHS of in Equation (7.2) are defined by the equations given in Section 4.5 of Chapter 4. These equations are:

$$\alpha_i = \alpha_{vi}\alpha_{ci}\,,\ \ \mathrm{i} = 1{,}2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7.3)$$

$$\alpha_{c1} = \alpha_{c2} = 1 - \phi \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7.4)$$

$$\alpha_{v1} = 1 - \frac{c_1}{R_{12}}\frac{\lambda_2^0}{\lambda_m^0}\,,\ \text{theoretical max. value of } c_1 = c_{\max 1} \approx 2\phi^2 \ldots\ldots(7.5)$$

$$\alpha_{v2} = 1 - c_2 R_{12}\frac{\lambda_1^0}{\lambda_m^0}\,,\ \text{theoretical max. value of } c_2 = c_{\max 2} \approx 2\phi^2 \ldots(7.6)$$

By substituting all relevant variables into the equations, the modified sets of transport equations can be said to model correctly the physics of flow, if a plot of λ_i^* and $\alpha_i\lambda_i^0$ against saturation gives two curves that almost overlap one another.

7.2.2. Analytical Approach II

In the second approach, Equations (7.1a) and (7.1b) can be rearranged in terms of the potential gradient as follows:

$$\frac{\partial\psi_1}{\partial x} = -\frac{v_1}{\lambda_1^0} - \left(\frac{1-\alpha_1}{2}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g.\sin\Theta\right] \ldots\ldots\ldots\ldots(7.7a)$$

and

$$\frac{\partial\psi_2}{\partial x} = -\frac{v_2}{\lambda_2^0} + \left(\frac{1-\alpha_2}{2R_{12}}\right)\left[\frac{\partial P_c}{\partial x} - (\rho_1 - \rho_2 R_{12})g.\sin\Theta\right] \ldots\ldots\ldots\ldots(7.7b)$$

where the term, $\partial P_c/\partial x$, in the RHS of Equations 7.7a and 7.7b is defined by:

$$\frac{\partial P_c}{\partial x} = \frac{\partial P_c}{\partial S}.\frac{\partial S}{\partial x} = R_{12}\frac{\partial P_2}{\partial x} - \frac{\partial P_1}{\partial x} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7.8)$$

and

$$v_i = \frac{q_i}{A} = f_i.v \qquad \mathrm{i} = 1{,}2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(7.9)$$

Again, all the variables on the RHS of Equations (7.7a), (7.7b) and (7.8) are measurable directly or derivable from experimental data. By substituting the values of the variables, the potential gradient distribution with distance can be calculated. These calculated values can then be fitted polynomially as a function of distance, and then integrated to obtain the potential distribution. The modified sets of transport equations can be said to model correctly the physics of flow, if the potential gradient or potential values, computed with the resulting integrated polynomial function, compare favourably, within a margin of error, with the measured potential gradient or potential distribution for all sets of experiments conducted.

7.2.3. Validation with Experimental Data

7.2.3.1. Data Group A: Bentsen (1998b and 1997)

Data group A is comprised of the results presented by Bentsen (1997 and 1998b). These unstabilized displacement experimental results are a combination of experimental data presented by Sarma and Bentsen (1989a) and Bentsen and Manai (1991). This data set includes core (sand pack) and fluids data, cocurrent and countercurrent effective and relative permeability data, and capillary pressure-saturation data, as shown in Table 7.1, Figures 7.1a and 7.1b, and Figures 7.2 respectively. Other data include fractional flow data, pressure (potential) gradient data, and saturation-distance profile data at an unsteady state displacement time of 1020 seconds, which are shown in Figures 7.3, 7.4 and 7.5, respectively. It should be noted that the pressure gradient data equals the potential gradient data because the experiments were conducted without the influence of gravity (that is, displacements in a horizontal core).

These experimental data have been smoothed from the original data using the least-square fitting technique. Ideally, the data from the experimental sources (SSCO, USCO and SSCC data) should be acquired using the same sand-fluid systems. Because of the rarity of such ideal data sets, these data sources were used. This can be accepted because the same sand and similar fluids were used in the experimental sets.

Bentsen (1998a) presented cocurrent and countercurrent effective mobility data. These effective mobility data were converted to effective permeability data (Figure 7.1a) and then to relative permeability data (Figure 7.1b) using the definition of effective mobility, which gives the relationship between effective mobility and relative permeability. This definition is given in the "Greek Letter" section of the nomenclature with the reference permeability taken as the effective permeability to each phase at the residual saturation of the other phase (that is, at end point effective permeability values).

The saturation values presented in all the figures are normalized saturation values computed from the original non-normalized saturation values presented by Bentsen. Also, the fitted effective and relative permeability coefficients and capillary pressure coefficients were determined by normalizing the saturation data presented by Bentsen (1998b) and by fitting the effective permeability and relative permeability data and capillary pressure data against normalized saturation. This is necessary in order for the data set to conform to the variables defined in the modified set of equations. Figures 7.1a, 7.1b and 7.2 show the fitted effective permeability, relative permeability and the capillary pressure data, respectively.

Details of the experimental procedure and equipment descriptions can be found in Bentsen (1998b and 1997), Bentsen and Manai (1991) and Sarma and Bentsen (1989a). Relative standard errors for all collected data remain less than 8%. So, the data set can be used with some degree of confidence.

Table 7.1: Properties of Core and Fluids and Other Displacement Data for Data Group A

Core measurement length	0.985 m
Core width/thickness	0.05 m
Core height	0.011 m
Core cross-sectional area	0.000550 m^2
Core inclination, Θ	0.00 degree
Core pore volume	172.3 x 10^{-6} m^3
Sand type	Ottawa silicate sand (80-120 mesh)
Average porosity	31.8%
Absolute permeability to wetting phase	17.17 x 10^{-12} m^2
Cocurrent effective perm. to non-wetting at S_{1i},	10.38 x 10^{-12} m^2
Countercurrent effec. perm. to non-wetting at S_{1i},	7.197 x 10^{-12} m^2
Cocurrent effective perm. to wetting at S_{2r},	5.114 x 10^{-12} m^2
Countercurrent effec. perm. to wetting at S_{2r},	3.516 x 10^{-12} m^2
Wetting phase (water) viscosity at 21^0C	0.001 Pa.s
Wetting phase (water) density at 21^0C	998.2 Kg/m^3
Non-wetting phase (LAGO) viscosity at 21^0C	0.0047 Pa.s
Non-wetting phase (LAGO) density at 21^0C	796.3 Kg/m^3
Initial wetting phase saturation	7%
Residual non-wetting phase saturation	10%
Flow type	SSCO, USCO and SSCC
Area under capillary pressure-saturation curve, A_c	1356.6 Pa
Capillary pressure-saturation data (curve)	See Figures 7.2
Capillary number, N_c	0.0581
Gravity term, N_g	0.000
Cocurrent end point mobility ratio, M_{rco}	~2.3
Countercurrent end point mobility ratio, M_{rcc}	~2.3
Effective permeability/Relative permeability data (curve)	See Figures 7.1a and 7.1b
Fractional flow data (curve) at 1020 sec. (USCO)	See Figure 7.3
Pressure/potential gradient data (curve) at 1020 sec. (USCO)	See Figure 7.4
Saturation-distance flow data (curve) at 1020 sec. (USCO)	See Figure 7.5
Value of a for $R_{12} = 1 - a(1 - S)$ determination	0.05
Average total displacement rate (USCO) , q	6.667 x 10^{-8} m^3/s
Average total displacement velocity (USCO), v	12,121.82 x 10^{-8} m/s
Total displacement rate and velocity (SSCO and SSCC)	Varied

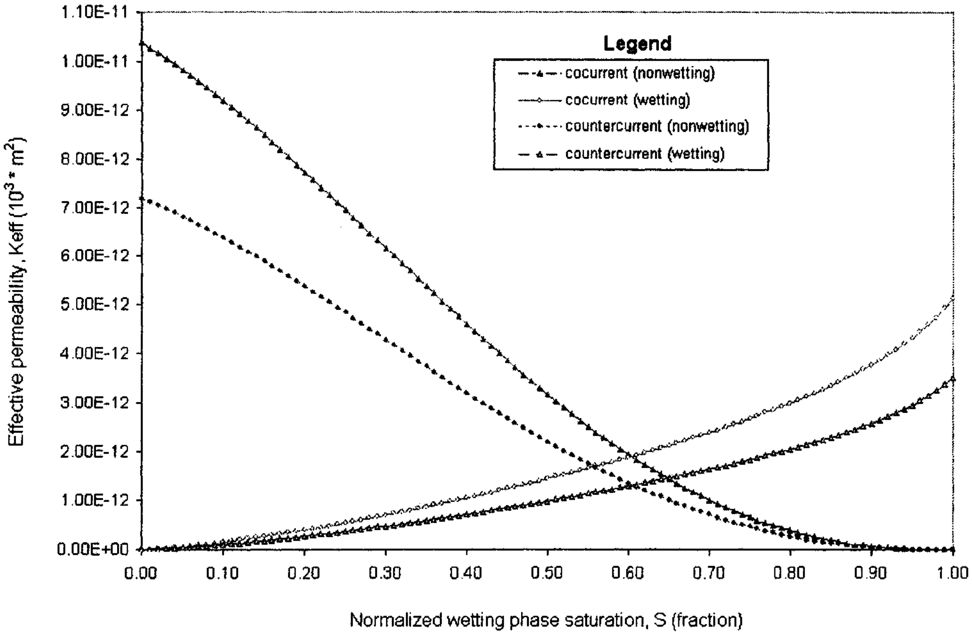


Figure 7.1a: Effective permeability curves - Data Group A

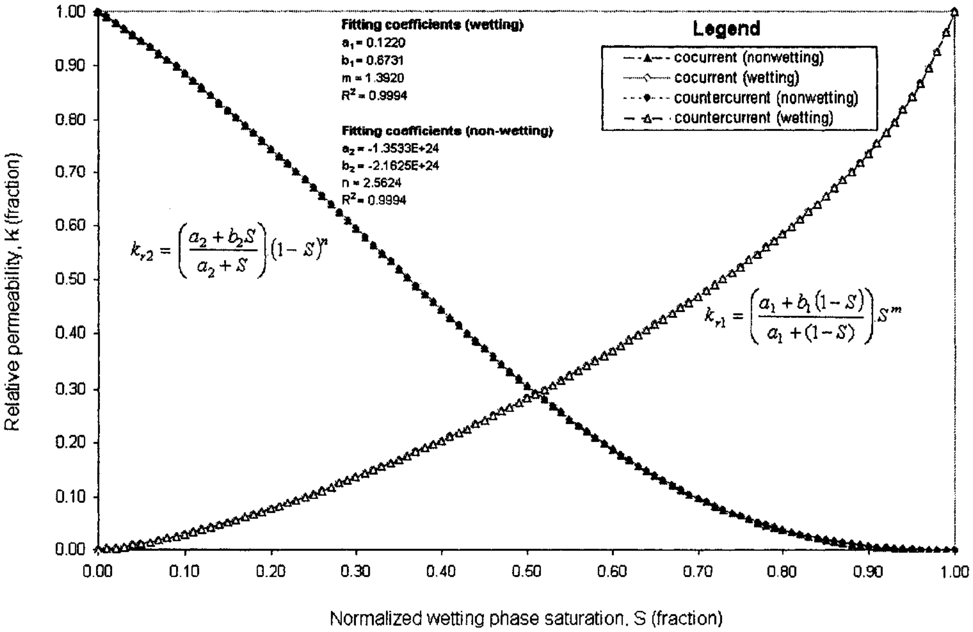


Figure 7.1b: Relative permeability curves - Data Group A

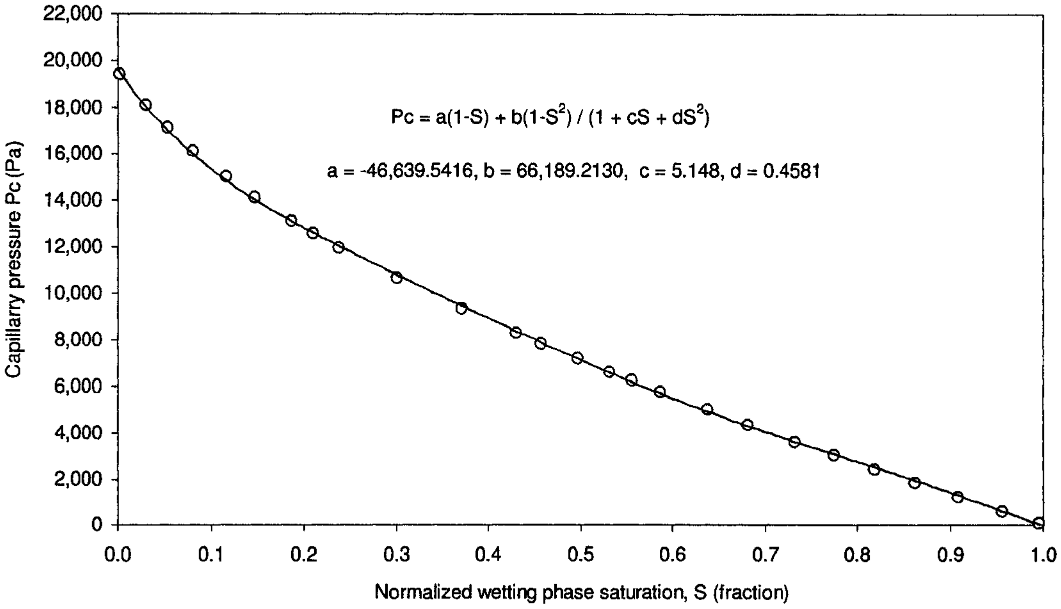


Figure 7.2: Capillary pressure-saturation curve - Data Group A

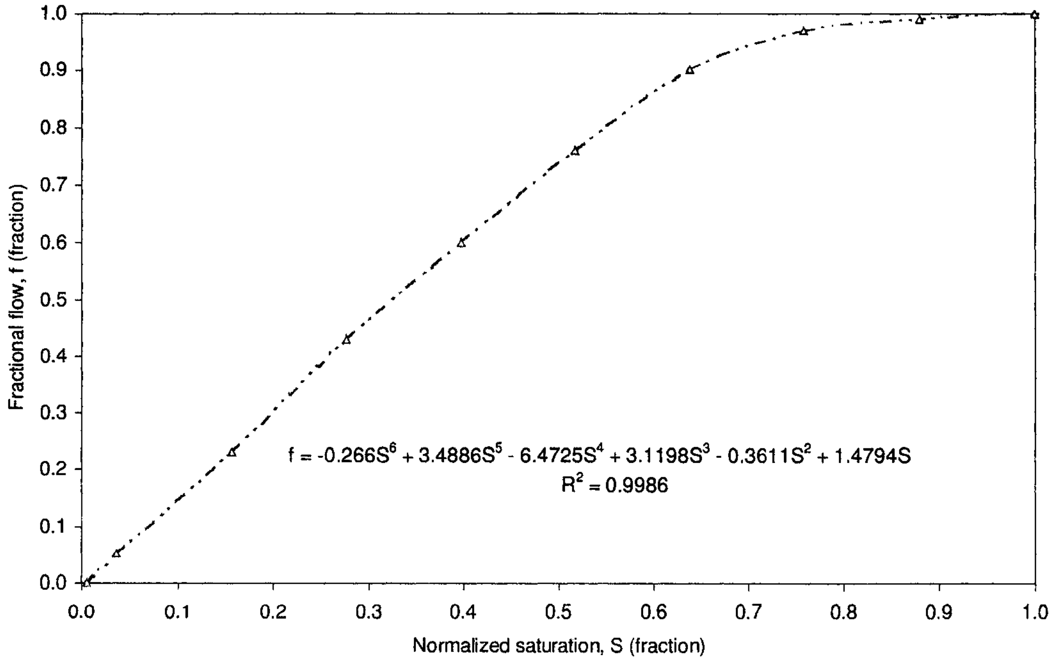


Figure 7.3: USCO fractional flow profile at 1020 sec. - Data Group A

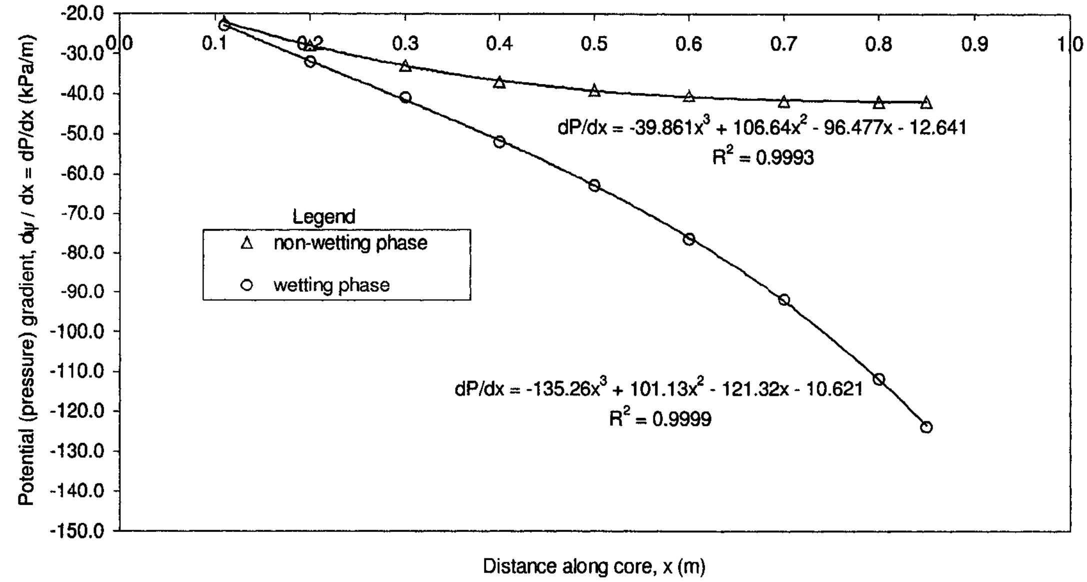


Figure 7.4: USCO pressure gradient profiles at 1020 sec. (Data Group A: Bentsen (1998a and 1997))

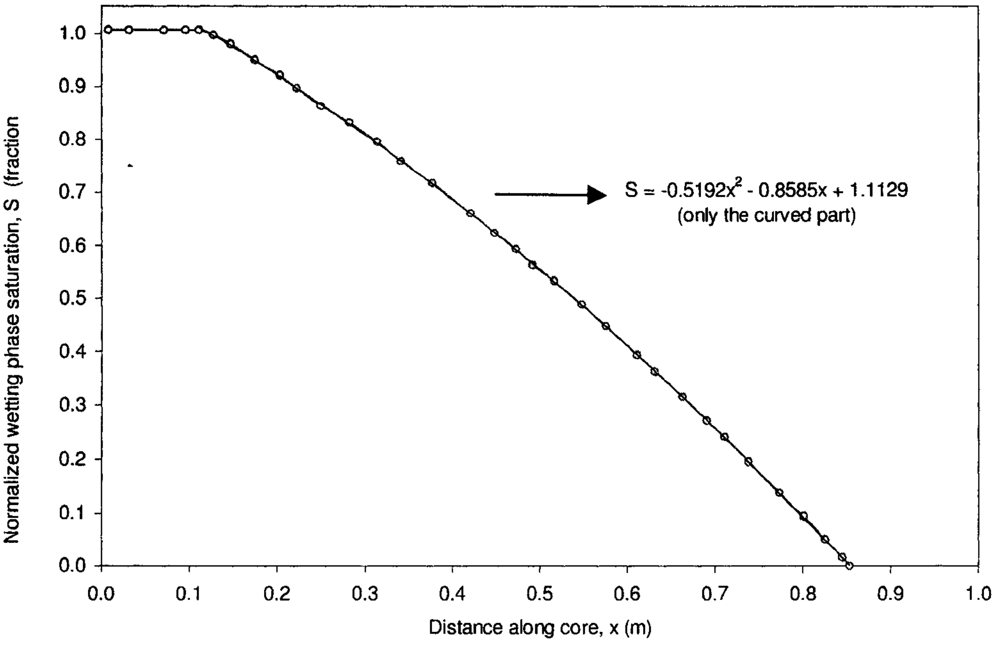


Figure 7.5: USCO saturation-distance profile at 1020 sec. - Data Group A

7.2.3.1.1. Discussion of Data Group A Results (Approach I)

Figure 7.1a shows the cocurrent and countercurrent effective permeability curves without any interfacial coupling correction applied. From the figure, it can be seen that the cocurrent effective permeability curves are much higher than the countercurrent effective permeability curves. Using the first experimental approach, the capillary coupling coefficient at each saturation value was computed using Equation (7.4) and the results were applied to Figure 7.1a to generate the plot in Figure 7.6. From Figure 7.6, it can be seen that the cocurrent effective permeability curves, with the capillary coupling applied, are almost the same as the countercurrent effective permeability curves. That is, the corrected curves almost overlap the countercurrent effective permeability curves. This shows that the introduction of the capillary coupling term leads to a much better description of flow.

Again, applying the computed viscous coupling coefficients at each saturation value, the curves shown in Figure 7.7 were generated using Equations (7.5) and (7.6). As can be seen in Figure 7.7, both the wetting and nonwetting effective permeability curves shifted slightly when the viscous coupling term was applied. The values of c_1 and c_2 used in the computation of the viscous coupling terms were taken to be equal to $2\phi^2$; that is, $c_1 = c_2 = c_{\max} = 2\phi^2$, the maximum value of c theoretically derived in Section 4.5.1.1 and also reported in Equations (7.5) and (7.6). These results indicate that viscous coupling cannot be used to explain the larger differences in the values of cocurrent and countercurrent effective permeability, depicted in Figure (7.1a), although it is yet to be shown if the theoretical value of $c_1 = c_2 = c_{\max} = 2\phi^2$ is correct. The validity of this expression is shown below.

Figure 7.8 shows the generated curves when both the viscous and capillary coupling corrections were applied. The cocurrent and countercurrent curves nearly overlap with the exception of the slight differences at higher values of the effective permeability for the non-wetting phase. This might be due to measurement error close to the end points. This might also be due to the fact that the core sample might have been disturbed during continuous injection and desaturation. That is, obtaining exactly the same injection conditions and sample is very difficult if not impossible, even in replicated runs. Achieving the desired (best) overlapping result was achieved by means of an interactive approach. The most appropriate experimental expression for c_1 and c_2 was found to be approximately $0.001\phi^2$. That is, the expression, $c_1 = c_2 \approx 2\phi^2$, did not give a good result, but by iteratively reducing the value from 2 to 0.001, a good match was found.

Using the maximum theoretical expression leads to a slight under prediction of the cocurrent effective permeability values for both the wetting and non-wetting phases, as shown in Figure 7.9. As a result of this, the correct expression, obtained from the experimental data, that is, $c_1 = c_2 \approx 0.001\phi^2$, was used to regenerate the curves in Figure 7.7. The regenerated curves are shown in Figure 7.10. The figure gives the correct profiles when the viscous coupling correction is applied to the cocurrent effective permeability. When compared with Figure 7.7, Figure 7.10 gives a lower separation of curves (wetting and non-wetting) because the error has been eliminated through the use of a correction factor obtained from experimental sources. Hence, it can be inferred that capillary coupling has a much large impact than does viscous coupling, and incorporating both types of coupling leads to a much better modeling of the physics of flow. Also, the results above show that neglecting viscous coupling introduces insignificant error into the transport equations.

Figure 7.1b shows the cocurrent and countercurrent effective permeability curves when they have been normalized to end point effective permeability values (that, is the relative permeability data). From the figure, it can be seen that both the cocurrent and countercurrent curves overlap. This a very important observation that has been noted by some earlier investigators who argue that what matters most in relative permeability data is the end point effective permeability values and not the intermediate values. The results presented in Figure 7.1b further reinforce this idea.

This observation also shows that either the conventional cocurrent or the countercurrent relative permeability values can be determined experimentally and used in reservoir simulation provided the data are normalized to the end points. Furthermore, if the conventional experimentally determined countercurrent effective permeability values are used in simulating a countercurrent flow recovery process, then the recovery prediction would be correct. On the other hand, if the conventional experimentally determined countercurrent effective permeability values are used in simulating a cocurrent flow recovery process, then the countercurrent effective permeability values would have to be divided by the interfacial coupling parameter, $\alpha_i = \alpha_{ci}.\alpha_{vi}$, in order to obtain an accurate recovery prediction. Also, the effective permeability values would have to be multiplied by the interfacial coupling parameter if the conventional experimentally determined cocurrent effective permeability values are used in the numerical simulation of a countercurrent recovery process. Also, of importance is the end-point mobility ratio, M_r. To calculate, M_r, end point effective and relative permeability values to each phase need to be known. In the analysis above, the M_r values for cocurrent and countercurrent data are approximately the same. In reality, it is extremely difficult to conduct experiments in which the cocurrent and countercurrent experiments have the same end point mobility ratio value.

It should be noted that the hydrodynamic effect was also included ($R_{12} \neq 1$) in the computation of the effective and relative permeability data. Bentsen (1998a) had earlier on showed that the non-inclusion of this effect does not introduce any significant error into the computation of relative or effective permeability. It can be concluded that if one neglects interfacial coupling effects (capillary and viscous) in the formulation of transport equations, then the conventional Darcy's equation must be written in terms of relative permeability data that use the end point effective permeability values as the reference permeability. However, if the transport equations are written in terms of effective permeability, the interfacial coupling effects should be included in the values determined using the expressions presented in this research; that is, Equations (7.1) to (7.6).

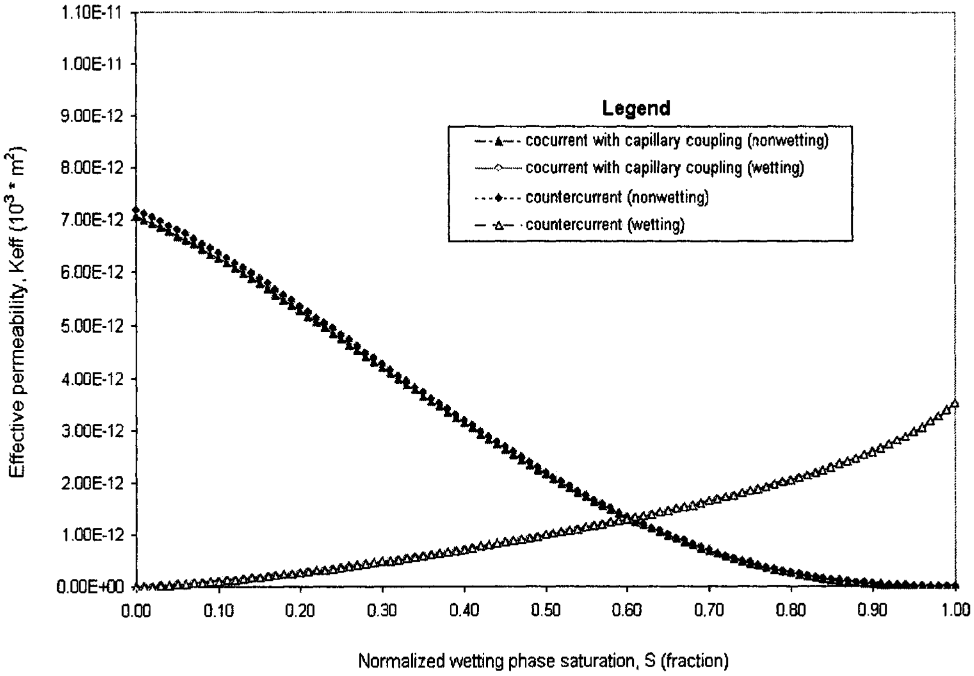


Figure 7.6: Effective permeability curves with only capillary coupling correction applied to cocurrent flow - Data Group A

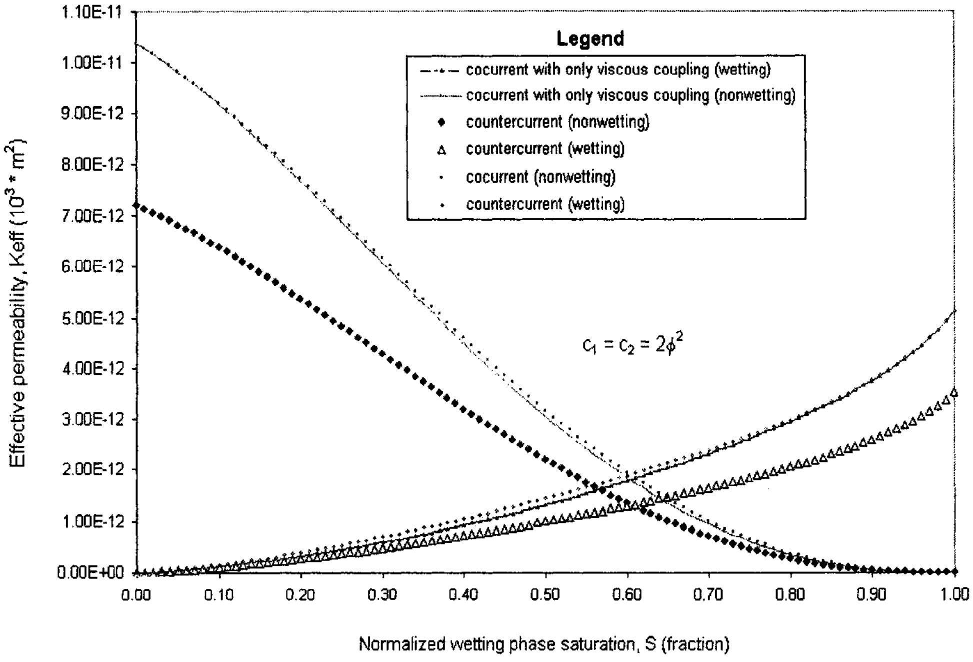


Figure 7.7: Effective permeability curves with only viscous coupling (theoretical) correction applied to cocurrent flow - Data Group A

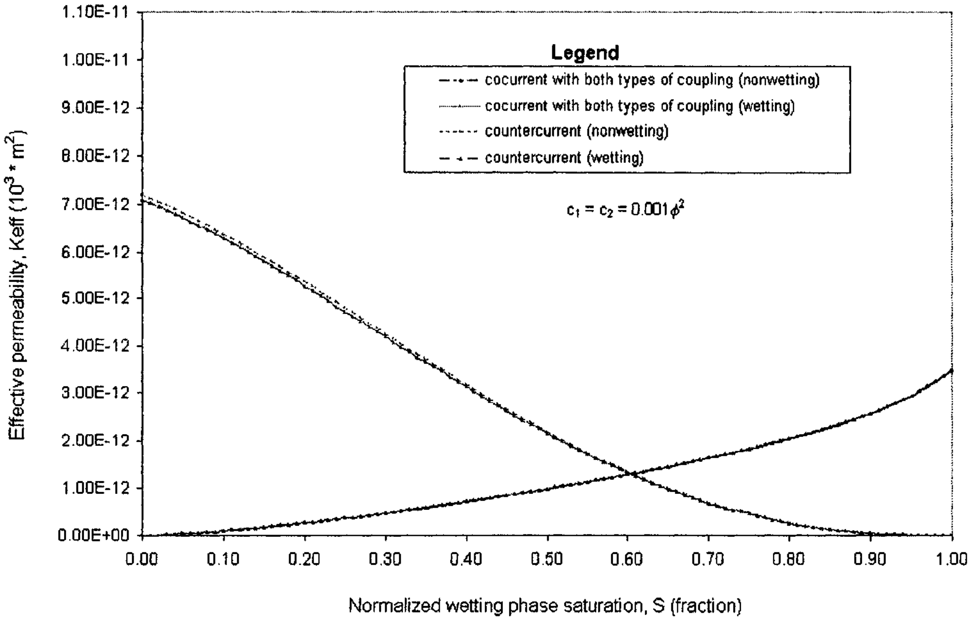


Figure 7.8: Effective permeability curves with both viscous (experimental) and capillary coupling corrections applied to cocurrent flow - Data Group A

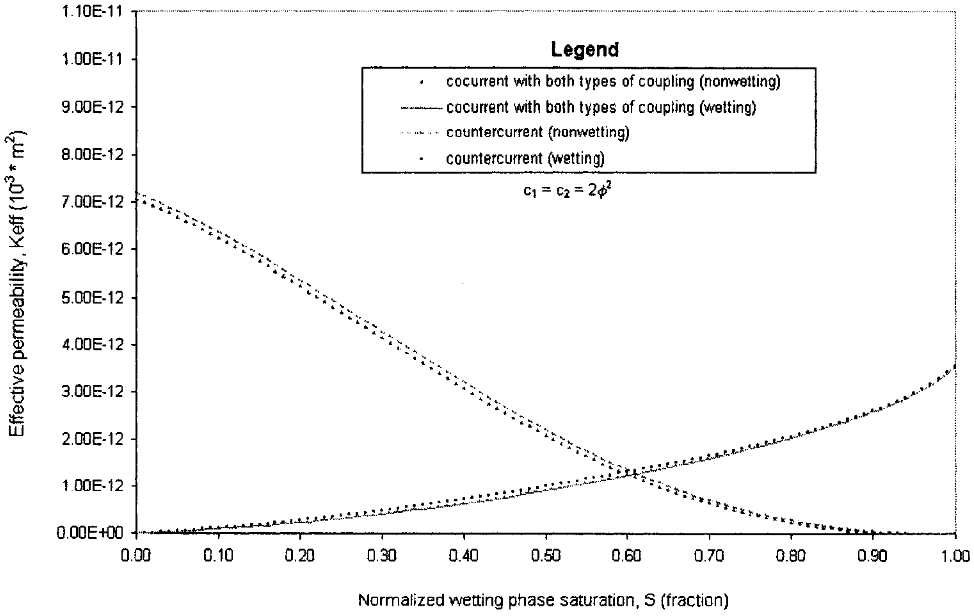


Figure 7.9: Effective permeability curves with both viscous (theoretical) and capillary coupling corrections applied to cocurrent flow - Data Group A

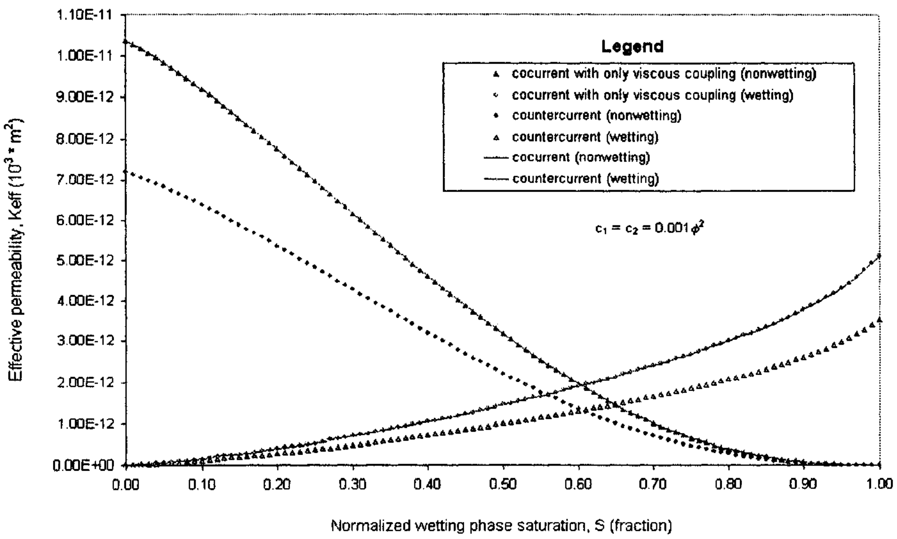


Figure 7.10: Effective permeability curves with only viscous coupling (experimental) correction applied to cocurrent flow - Data Group A

7.2.3.1.2. Discussion of Data Group A Results (Approach II)

In the second approach, the potential (pressure) gradients were computed using Equations (7.7a) and (7.7b). All the variables on the RHS of Equations (7.7a) and (7.7b) can be computed directly from the data in Table 7.1, Figures 7.1b and 7.4, and Equations (7.3) to (7.6). The values of $\partial P_c/\partial x$ were computed using Equation (7.8) and the data given in Figures 7.4 and 7.5, while velocity values were computed with Equation (7.9). The computed potential (pressure) gradients are plotted along with the directly measured pressure (potential) gradients of Figure 7.4 in Figure 7.11.

The computed profiles in both phases were only compared with experimental data up to about 0.75 m of the core, which is the point at which the last pressure sensor on the core was located. In the original experiments presented by Sarma and Bentsen (1989a), pressure data beyond this point were extrapolated and as a result were not used in the comparison. From Figure 7.11, it can be seen that, both the computed and the measured wetting and nonwetting phases potential (pressure) gradients compared favorably along the core length, within the limits of experimental error. The best results for both phases phase were obtained at the middle of the core, while those close to the inlet and outlet ends show slight variations. The slight deviations of the wetting and nonwetting computed pressure gradients from the wetting and nonwetting measured pressure gradients, respectively, close to the ends is due to experimental error. But agreement between the pressure gradient profiles is close enough and can be accepted as being fairly good results.

One can conclude that the modified set of transport equations give a good description of flow not only in terms of relative permeability and effective permeability but also in terms of pressure (potential) gradient prediction. The experimental data presented in this section were acquired without the effect of gravity. Data Group C (Bourbiaux and Kalaydjian (1990)) analyzed later was conducted under the effect of gravity and capillarity.

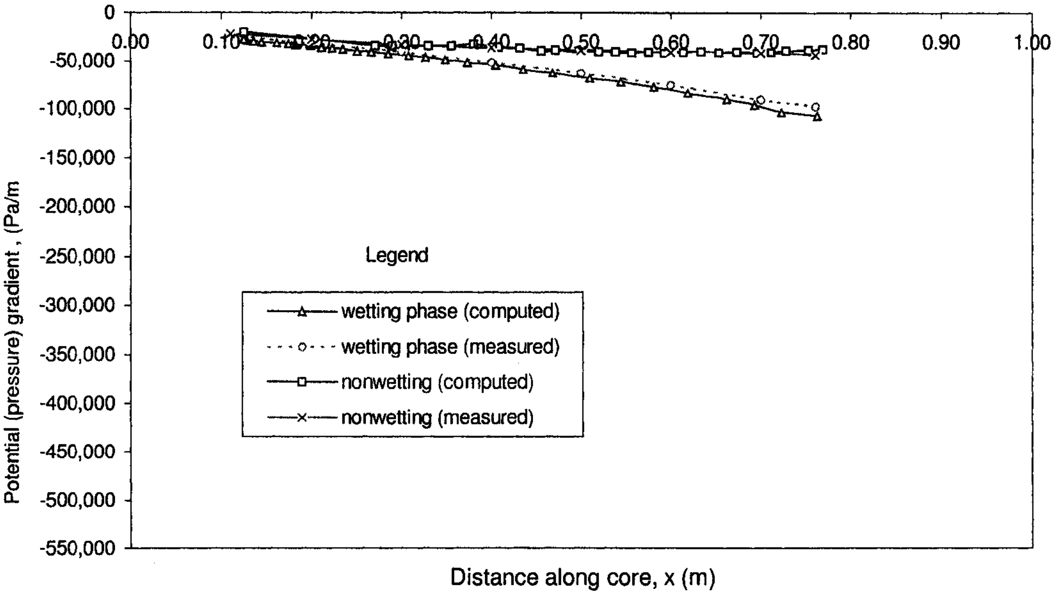


Figure 7.11: Comparison of computed and measured potential (pressure) gradient curves - Data Group A

7.2.3.2. Data Group B: Acquired Laboratory Data

Data group B consists of experimental data that were acquired as described in Chapter 6. These include SSCO and USCO (Type II) experimental data. The USCO (Type I) experimental data were not analyzed and are not presented here because the quality of the data is poor. The data for Type II experiments are shown graphically in Appendix E. The raw frequency measurements (Figure E1) look fairly good but the quality of the pressure measurements is not good. The wetting phase pressure measurements in Figure E2 exhibit positive slopes, which is not good for analysis purposes. This problem is likely partially due to mechanical or electrical damage of the pressure transducer located at the last measurement point on the core, where the pressure value seems to remain constant without changing over time.

In addition, in the wetting and nonwetting phases, the pressure profiles reduce over time, then increase again and afterwards decrease. This up and down behaviour was thought to be due to compressibility effects associated with the core, changes in the hydrophilicity of the water-wet membranes and measurement error of the transducers. Normally, for a complete set of experiments, USCO experiments are conducted before SSCO experiments on the same sand-fluid system. After noticing the problems associated with the pressure measurements, Type I experiments were discontinued and a new set of USCO and SSCO (Type II) was collected. For analysis purposes, USCO and SSCO experimental data on the same sand-fluid system is preferred, if it can be acquired. The quality of the Type II - USCO pressure profiles is fairly good for analysis purposes especially in the nonwetting phase. The wetting phase pressure measurements for Type II - USCO run also exhibited positive slopes.

Table 7.2: Properties of Core and Fluids and Other Displacement Data for Data Group B at Laboratory Conditions

Core total length	0.418 m
Core saturation measurement length	0.180 m
Core pressure measurement length	0.265 m
Core width/thickness	0.05 m
Core height	0.01 m
Core cross-sectional area	0.00050 m^2
Core inclination, Θ	0.00 degree
Core pore volume (0 to 0.18 m)	27.09 x 10^{-6} m^3
Sand type	Ottawa silicate sand (80-120 mesh)
Average porosity	30.10%
Absolute permeability to wetting phase	25.746 x 10^{-12} m^2
Effective permeability to non-wetting phase at S_{1i},	22.547 x 10^{-12} m^2
Effective Permeability to wetting phase at S_{2r},	12.317 x 10^{-12} m^2
Wetting phase (water) viscosity	0.001000 Pa.s
Wetting phase (water) density	975 Kg/m^3
Non-wetting phase (light mineral oil) viscosity	0.025125 Pa.s
Non-wetting phase (light mineral oil) density	750 Kg/m^3
Initial wetting phase saturation	7.773%
Residual non-wetting phase saturation	7.112%
Flow type	SSCO and USCO (Type II)
Area under capillary pressure-saturation curve, A_c	~907.82 Pa
Capillary pressure-saturation data (curve)	See Figures 7.16c (From SSCO data)
Capillary number, N_c	0.99 > 0.01 (i.e. unstabilized)
Gravity term, N_g	0.000
Stability number, I_{sr}	0.0019 < π^2 (i.e. stable displacement)
End point mobility ratio, M_r	1.8297
Relative and Effective permeability data (curves)	See Figures 7.16a and 7.16b (From SSCO)
Saturation-distance flow data (curve) (USCO)	See Figures 7.12b and 7.12c
$R_{12} = 1 - a(1 - S)$ determination	0.00 (1.e. R_{12} = 1)
Average wetting phase Injection rate (USCO)	3.122382 x 10^{-8} m^3/s
Average wetting phase Injection velocity (USCO)	6.244764 x 10^{-5} m/s

Relevant core and fluid properties as well as displacement data, for the Type II data set, are shown in Table 7.2. Also, a summary of some of the experimental data/results is provided in Appendix D in tabular format. Results, relevant to the discussion presented in this section, in graphical format, are given in Figures 7.12 to 7.16. The raw frequency measurements (USCO and SSCO) were processed and converted to actual saturations as described in Section 6.4.3 of Chapter 6. Normalized saturation, S, values were then computed from these saturation measurements. The computed USCO saturation profiles are shown in Figure 7.12c while that for SSCO flow are shown in Figure 7.14c. Figure 7.12c depicts normalized wetting phase propagation fronts during the unsteady state experiments. A polynomial expression was used to fit saturation and frequency data (See Equation (6.4) and Figure 6.6) in the range that covers initial water saturation (7.77% at 85.046538 Hz) to irreducible water saturation (92.89% at 84.850492000 Hz). As a result of this, any frequency data recorded outside this range due to environmental or geometrical error, especially above 85.04 Hz would result in a negative saturation value if Equation (6.4) were used. To eliminate this error, any value recorded above 85.04 Hz is assumed to be approximately equal to a saturation value of 7.77%, the initial water saturation. This error might also be due partly to the non-uniform distribution of the initial water saturation. The saturation fronts move from near steep to flat as the injection continues from the initial unsteady condition to breakthrough to pseudo steady state and eventually to steady state conditions. It should be noted that the experimental data are not smooth due to local heterogeneity in the core and undesirable electromagnetic interference within the laboratory environment. These factors result in perturbation of the saturation profiles. Another source of perturbation is the fact that capillary forces are much higher than the viscous forces (N_c = 0.99). Consequently, small scale permeability variations result in large perturbations.

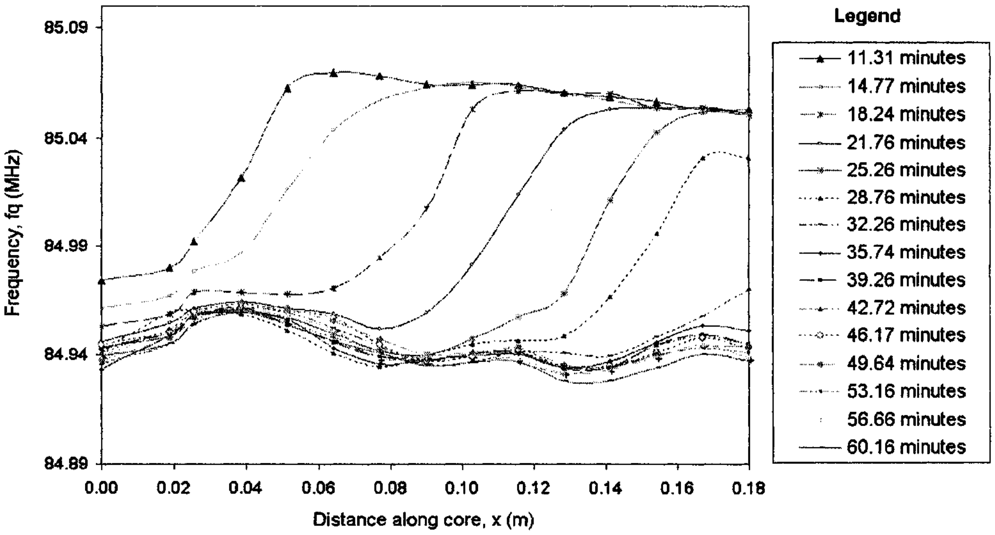


Figure 7.12a : Raw frequency measurements along the core (Data Group B: Type II - USCO)

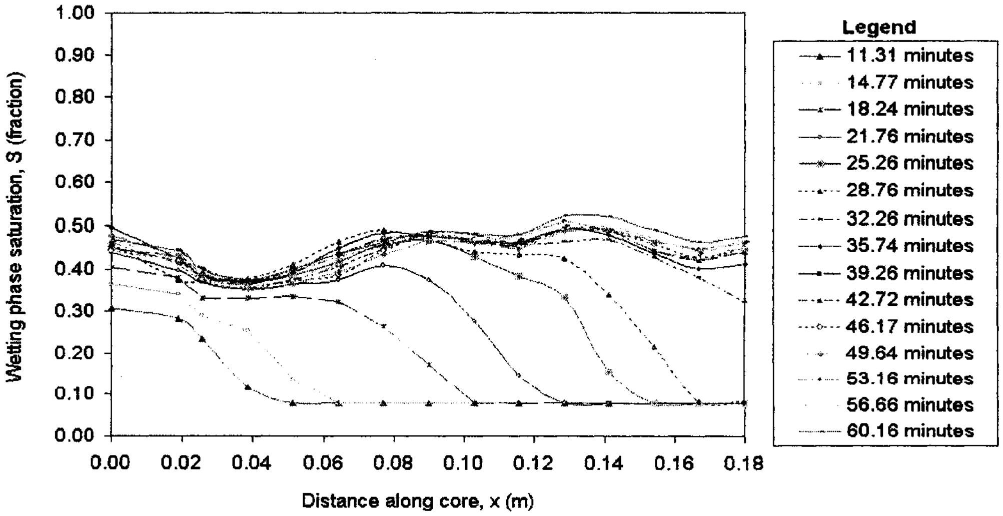


Figure 7.12b : Saturation-distance profiles (Data Group B: Type II - USCO)

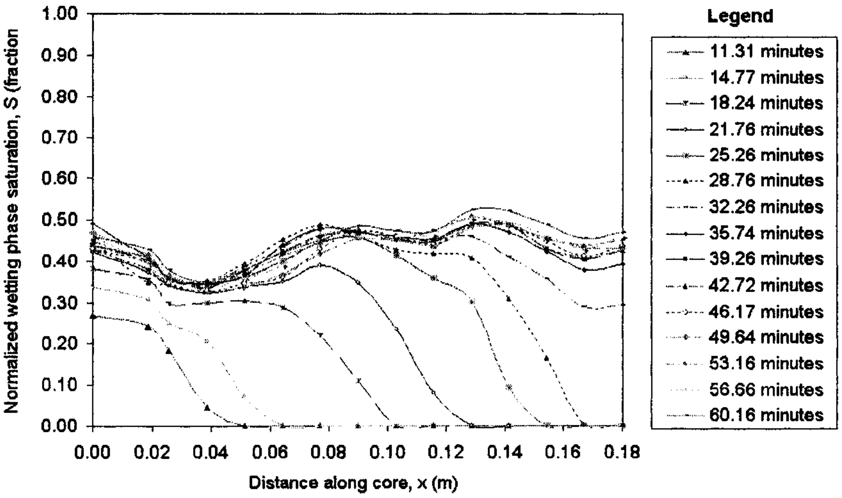


Figure 7.12c : Normalized saturation-distance profiles (Data Group B: Type II - USCO)

Under normal conditions, the normalized inlet saturation in Figure 7.12c should have risen gradually over time to 1. Why this did not occur might be due to selective entry at the inlet of the core and/or inlet end effect. Although the core was designed to eliminate the inlet end effect, it seems that the extra length at the core inlet to account for this is not enough due to inlet plate design. The experimental data are unstabilized but stable because the capillary number, N_c, is greater than 0.01 and the value I_{SR} is less than π^2 (see Table 7.2). Figures 7.13a and 7.13b show the pressure profiles in the wetting and non-wetting phases, respectively, during the unsteady state injection period. Pressure profiles in both phases were fitted with second order polynomial equations. The quality of the non-wetting phase pressure profiles is relatively good, but the quality of the wetting phase pressure profiles is poor due to the positive slopes exhibited by the profiles, as shown in Figure 7.13a. This problem is due to measurement error.

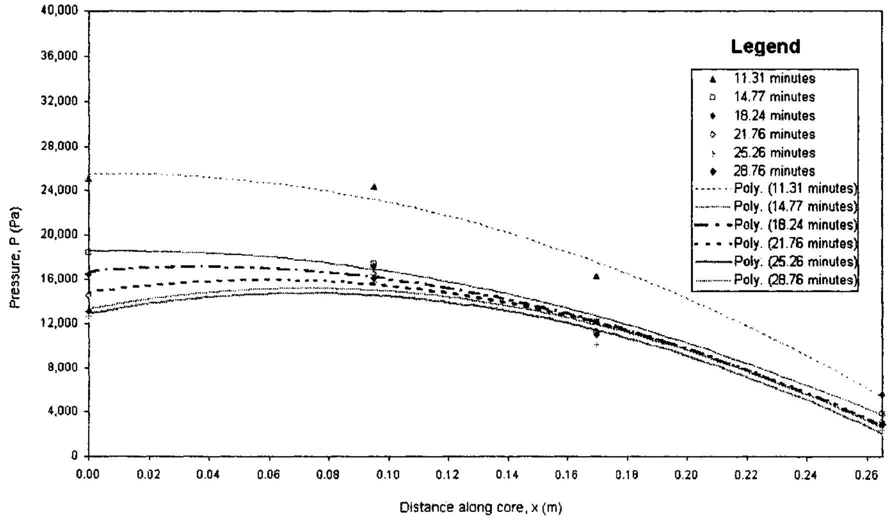


Figure 7.13a : Wetting phase pressure measurements along the core
(Data Group B: Type II - USCO)

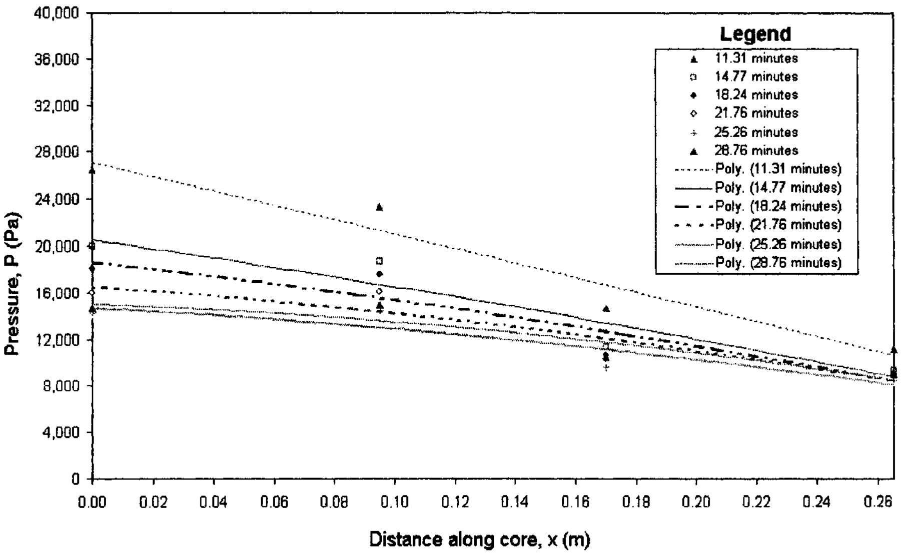


Figure 7.13b : Nonwetting phase pressure measurements along the core (Data Group B: Type II - USCO)

Figures 7.13c and 7.13d show the pressure data at the unsteady state time level of 21.76 minutes, the time level at which the analysis was carried out, for the wetting and nonwetting phases, respectively. The analysis was carried out at 21.76 minutes because the quality of the non-wetting pressure data at this time was better than at later times. The quality of the wetting phase pressure measurements in the USCO experiment (Figure 7.13c) is not very good due to the positive slope problem mentioned above. As a result of the above problem, only USCO nonwetting and SSCO wetting and nonwetting phase pressure measurements were used in the analysis. Also data Group B was acquired without the effect of gravity (horizontal flow); hence, the potential equals the pressure. Figure 7.13e shows the plot of USCO injection rate over time. The plot shows that the rate is fairly constant over the injection period.

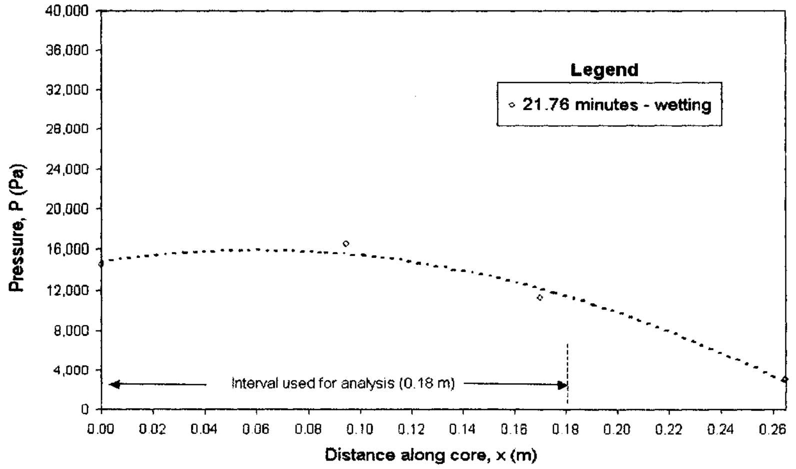


Figure 7.13c : Wetting phase pressure measurements along the core at time level of 21.76 minutes (Data Group B: Type II - USCO)

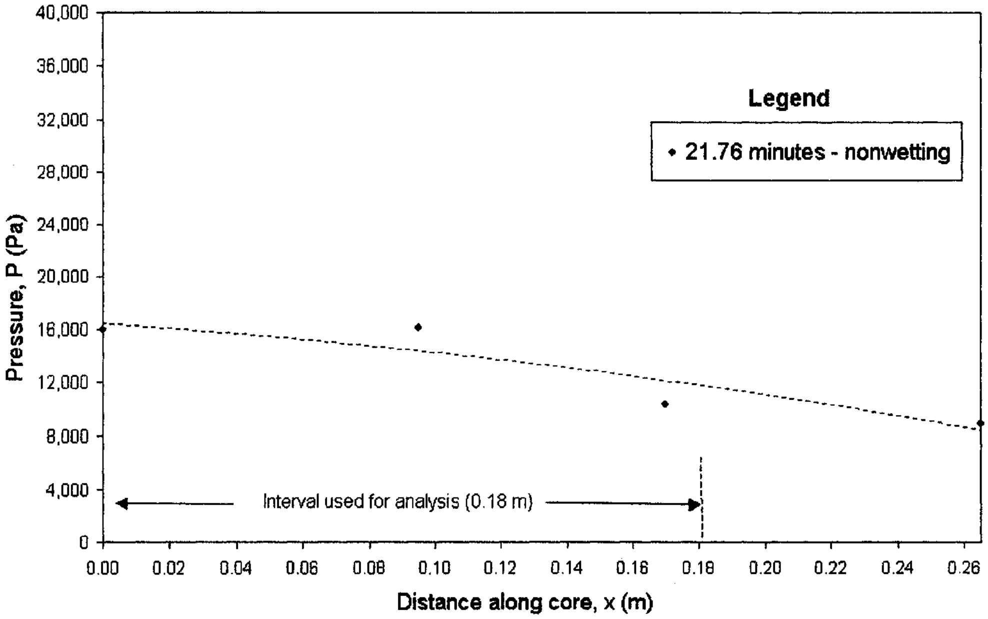


Figure 7.13d : Nonwetting phase pressure measurements along the core at time level of 21.76 minutes (Data Group B: Type II - USCO)

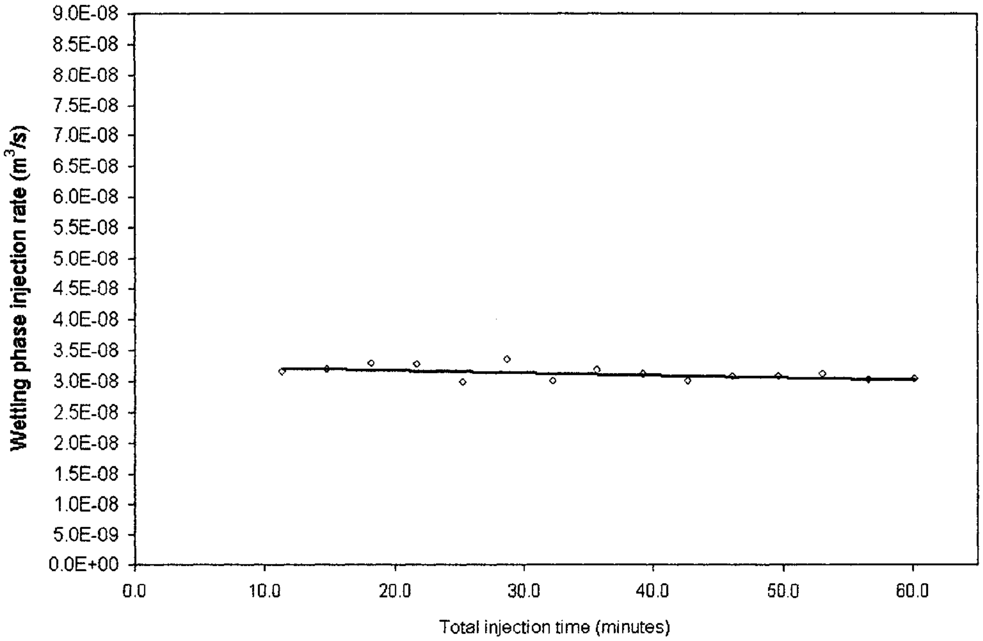


Figure 7.13e: Wetting phase injection rate profile (Data Group B: Type II - USCO)

Under ideal conditions, the original steady state frequency measurement curves shown in Figure 7.14a should be perfectly horizontal and parallel to one another because the saturation and hence frequency measurements should be the same along the core length. As shown in the figure, the curves are not perfectly horizontal due to geometrical and environmental effects as described in Section 6.4.3.2 of Chapter 6, but are fairly parallel. Geometrical and environmental corrections were applied to these original measurements as described in Section 6.4.3.2 of Chapter 6. The saturation-frequency calibration curve, for USCO saturation values, was based on these corrected measurements. The corrected SSCO saturation profiles and the normalized equivalents are shown in Figures 7.14b and 7.14c, respectively.

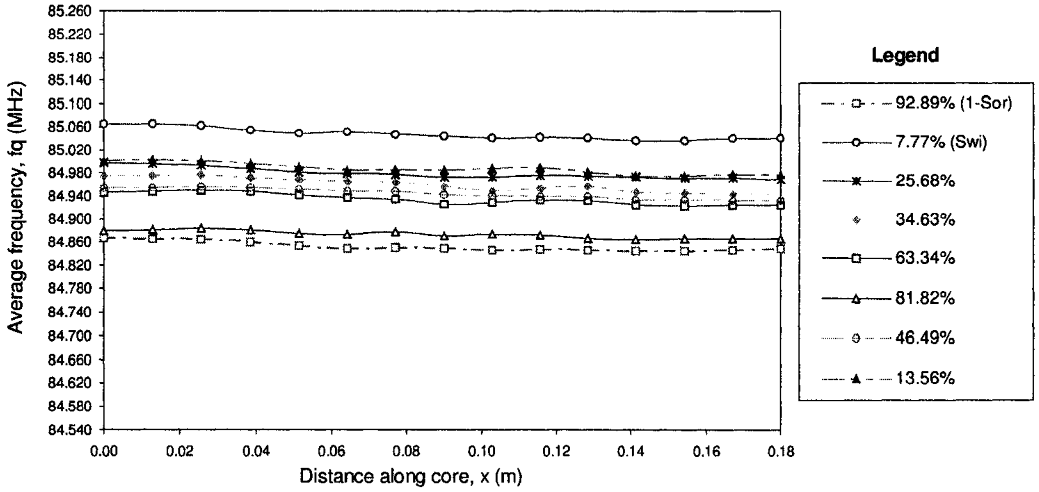


Figure 7.14a: Raw frequency measurements profiles along the core at steady state saturation levels (Data Group B:Type II - SSCO)

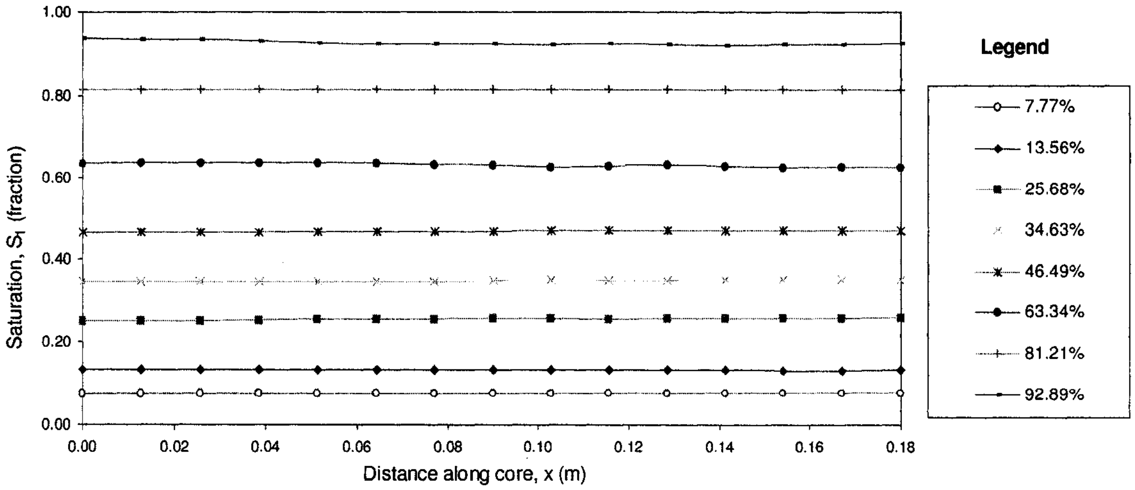


Figure 7.14b: Computed steady state saturation profiles along core holder after geometrical and environmental corrections (Data Group B: Type II - SSCO)

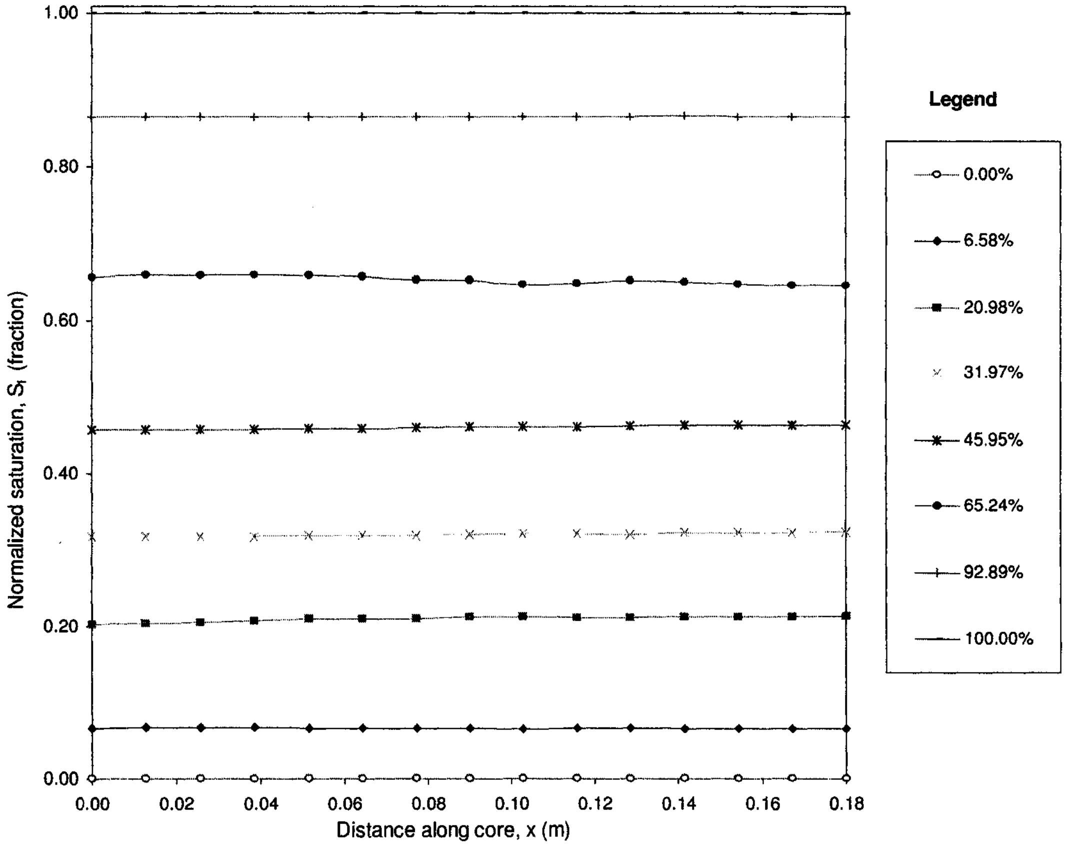


Figure 7.14c: Computed steady state normalized saturation profiles along core holder after geometrical and environmental corrections (Data Group B: Type II - SSCO)

The plot in Figure 7.15 shows a plot of pressure gradient ratio against $1-S$. The plot was used to estimate the value of R_{12} at every saturation point to account for the hydrodynamic effects reported by Manai (1991). From the plot shown in Figure 7.15, the value of a is 0.2739. The value of a is related to R_{12} by (Manai (1991) and Bentsen (1992)):

$$R_{12} = 1 - a(1-S) \quad \text{...} (7.10)$$

where a is the hydrodynamic fitting coefficient, a dimensionless parameter that is experimentally determined This value of a in Figure 7.15 is high and the correlation value, R^2, for the plot is poor (approximately -0.7884). To avoid the introduction of error into the analysis due to poor correlation, it was consequently decided to fix $R_{12} = 1$, that is, $a = 0$. This is not expected to introduce any significant error, especially in cocurrent flow (Bentsen (1998a)). Also, it should be noted that the nonwetting phase used by Manai is different from the nonwetting phase used in the present study.

The effective permeability values were calculated directly with the Darcy equation from SSCO experimental data because all the variables needed were measurable directly or indirectly. See Appendix D (Table D2) for other relevant data. The SSCO saturation data were also used in the determination of relative permeability values. The relative permeability is relative to the end point of each phase. The fitted relative permeability curves are shown in Figure 7.16a. The effective permeability data and the fitted effective permeability data are shown in Figure 7.16b. The fitted effective permeability curves shown in Figure 7.16b were fitted with the equations below (from Bentsen 1976):

$$k_{eff1} = \left(\frac{a_1 + b_1(1-S)}{a_1 + (1-S)} \right) . S^m \quad \text{.........(7.11)}$$

and

$$k_{eff2} = \left(\frac{a_2 + b_2 S}{a_2 + S} \right) . (1-S)^n \quad \text{.........(7.12)}$$

where a_1, b_1, a_2, b_2, m and n are the fitting coefficients. The fitted relative permeability data shown in Figure 7.16a were then generated from the fitted effective permeability data shown in Figure 7.16b using the relationship, $k_{ri} = k_{effi} / k_{refi}$. The term, k_{refi}, refers to the reference permeability, which is the end point of each phase.

Also, the capillary pressure data were computed at the inlet directly from SSCO experimental data by using the equations given below, which are based on the analysis given in Section 4.4:

$$P_c = P_2 - P_1 \quad \text{at the inlet, i.e. at } x = 0 \quad \text{.........(7.13)}$$

and

$$P_c = R_{12} P_2 - P_1 \quad \text{along the core, i.e. at } 0 < x \leq L \quad \text{.........(7.14)}$$

The capillary pressure data and the fitted data are also shown in Figure 7.16c. The capillary pressure data were fitted with the equation below:

$$P_c = aS^3 + bS^2 + cS + d \quad \text{.........(7.15)}$$

where a, b, c, and d are the fitting coefficients.

The analysis presented in the section below is based on only one of the analytical techniques (Analytical Approach II - Section 7.2.2) because of the non-availability of countercurrent flow data.

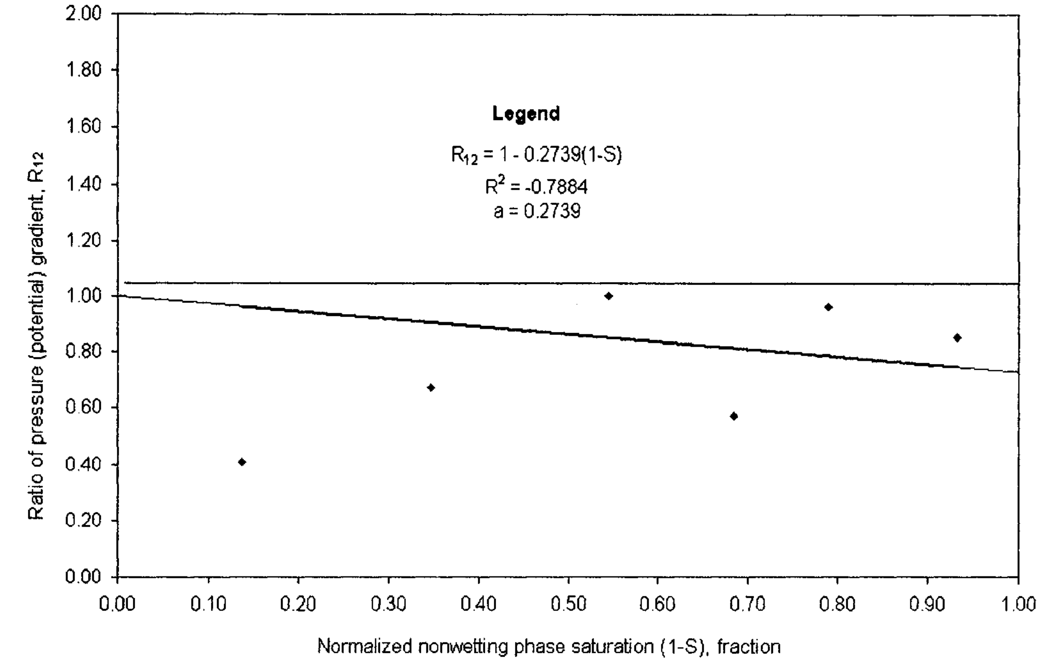


Figure 7.15 : Ratio of wetting to nonwetting potential (pressure) gradients at each saturation level (Data Group B: Type II - SSCO)

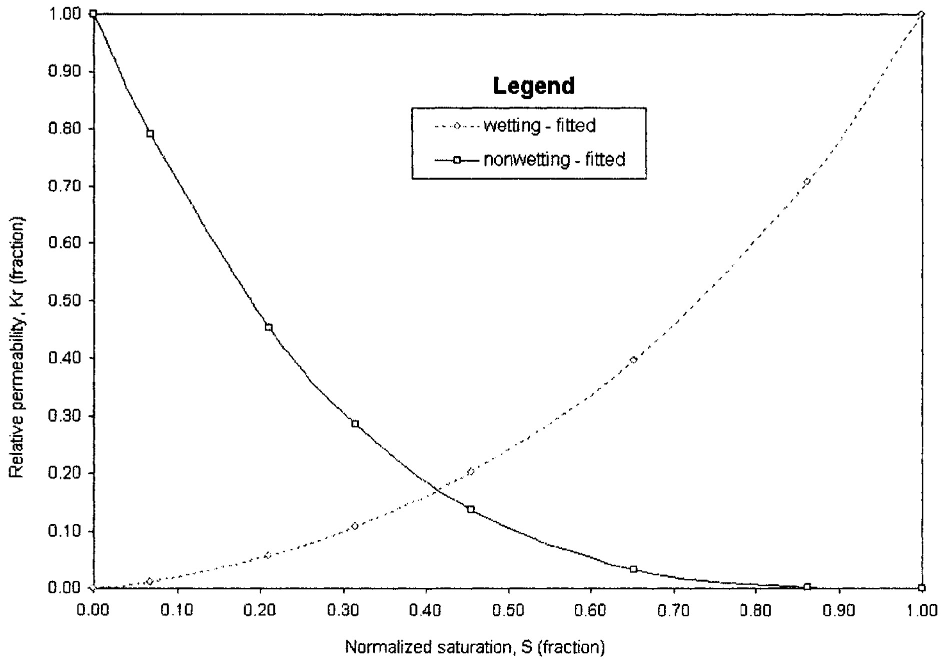


Figure 7.16a: Relative permeability curves (Data Group B: Type I I - SSCO)

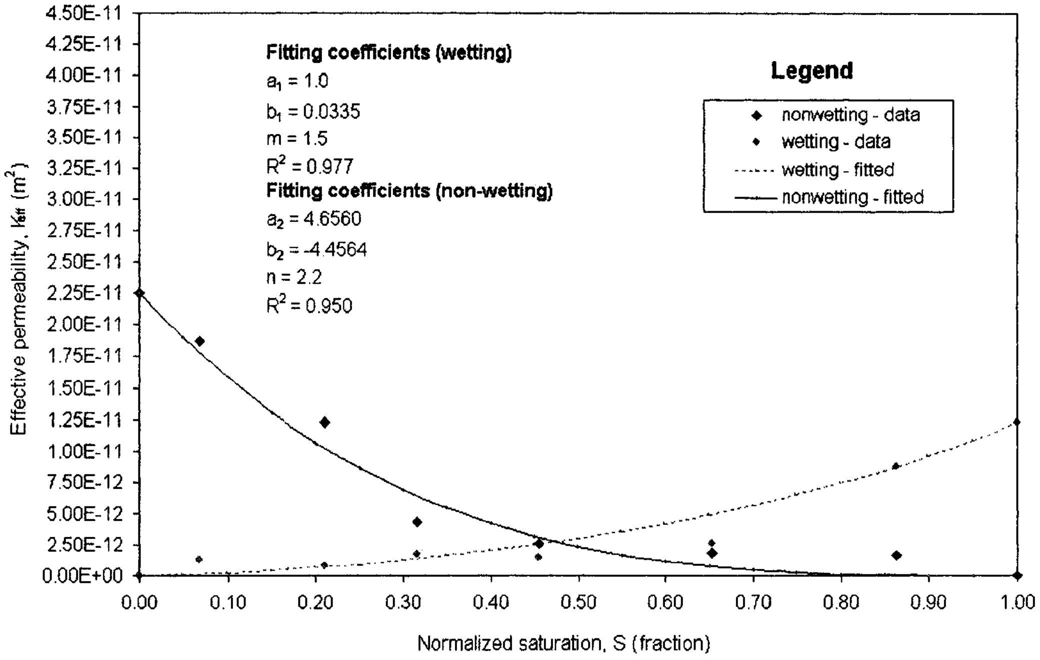


Figure 7.16b: Effective permeability curves (Data Group B: Type I I - SSCO)

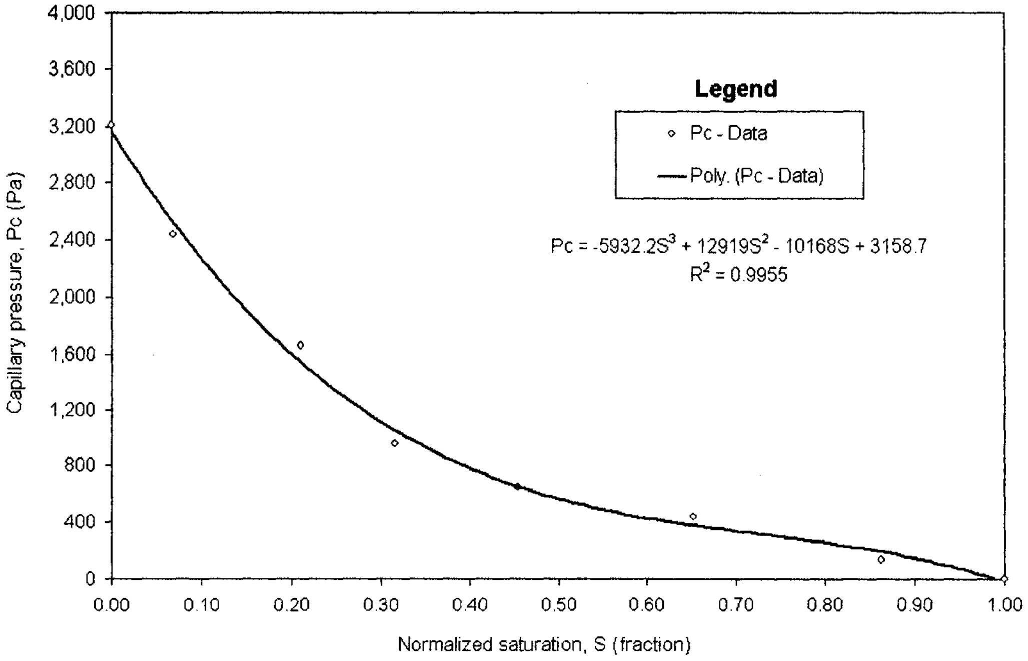


Figure 7.16c: Capillary-saturation curve (Data Group B: Type I I - SSCO)

7.2.3.2.1. Discussion of Data Group B Results (Approach II)

In this approach, experimental data at one of the unsteady state time levels is required. Data at a time level of 21.76 minutes were used. Pressure, flow and saturation data at this time interval are of much better quality and they cover much (more than half) of the core saturation measurement length. The saturation profile at this time level was also used as part of the data for analysis. Due to some perturbations in the saturation profiles as discussed earlier, the saturation data were smoothed as follows. Firstly, the saturation data between the core inlet and 0.06429 m of the core length were averaged-out to eliminate the positive slope in this interval. The averaged data and the raw data beyond this point were then combined to generate a parametric fitted curve of the saturation-distance profile at the above time level, as shown in Figure 7.16d.

Other relevant data required for the second approach include the values of λ_1^0, $\lambda_2^0, R_{12}, \alpha_1$ and α_2, v and $\partial P_c/\partial x$. The values of λ_1^0, $\lambda_2^0, R_{12}, \alpha_1$ and α_2 were obtained from the fitted SSCO experimental data while $\partial P_c/\partial x$ values were obtained from the fitted SSCO and USCO experimental data with Equation (7.8). The use of SSCO data, $\partial P_c/\partial S$, and USCO experimental data, $\partial S/\partial x$, to compute $\partial P_c/\partial x$ for the purpose of analysis as opposed to the use of only USCO experimental data, has been shown to be very reliable (Sarma (1988) and Ayub (2000)). It should be noted that the computation of α_1 and α_2 includes the effect of both viscous and capillary coupling. The computed fractional flow values, velocity and other relevant data were then substituted into Equation (7.7b) to compute pressure (potential) gradient profiles for the nonwetting phase at the time level of 21.76 minutes. The distance corresponding to a given potential gradient was obtained from the parametric fitted saturation profile shown in Figure 7.16d.

The measured nonwetting phase pressure profile shown in Figure 7.13b was fitted polynomially. The fitted curve was then differentiated with respect to distance along the core to obtain a smoothed estimate of the measured nonwetting phase pressure (potential) gradient. As stated earlier, the quality of the USCO wetting phase pressure measurements was not good enough. So, pressure gradients for the wetting phase data were not compared with experimental data. Figure 7.16e shows the plots of computed and smoothed measured pressure values for the non-wetting phase at the indicated time level. Due to uncertainty in the wetting phase pressure measurements in the region where only the nonwetting phase is flowing, the pressure gradient values were not plotted beyond this point, that is, the flood front. From the plots, the computed pressure gradient values compare favourably with the experimental results within the limits of experimental and curve fitting errors. Based on these results, one can conclude that the modified set of transport equations gives a favourable description of flow in cocurrent two-phase porous media flow, in which the effect of interfacial coupling is accounted for. In summary, the second analytical approach gives favourable results. But due to measurement errors in the USCO wetting phase pressure measurements, no comparison could be made.

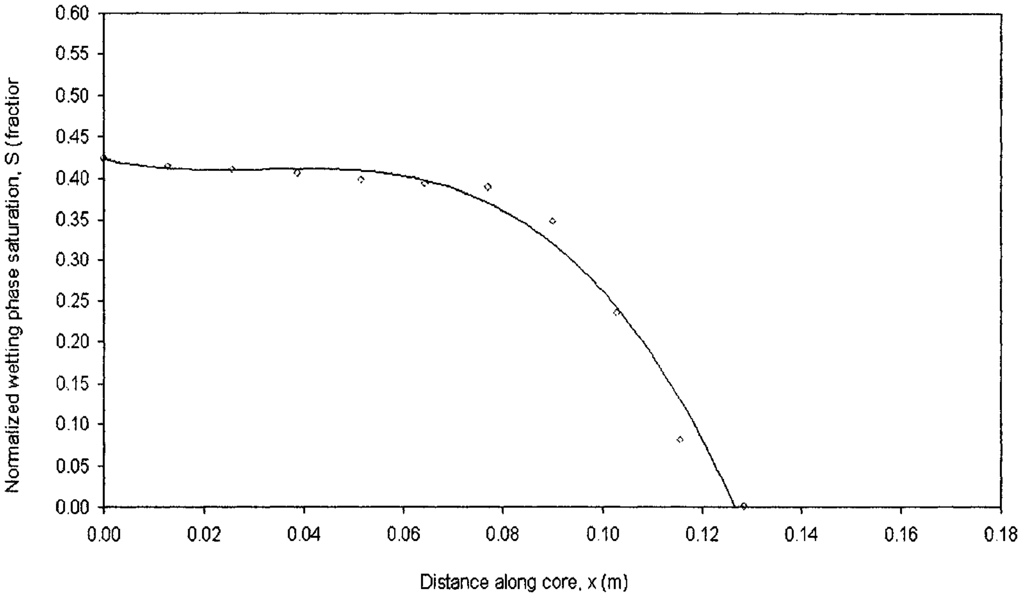


Figure 7.16d : Smoothed normalized saturation-distance profile at time level of 21.76 minutes (Data Group B: Type II - USCO)

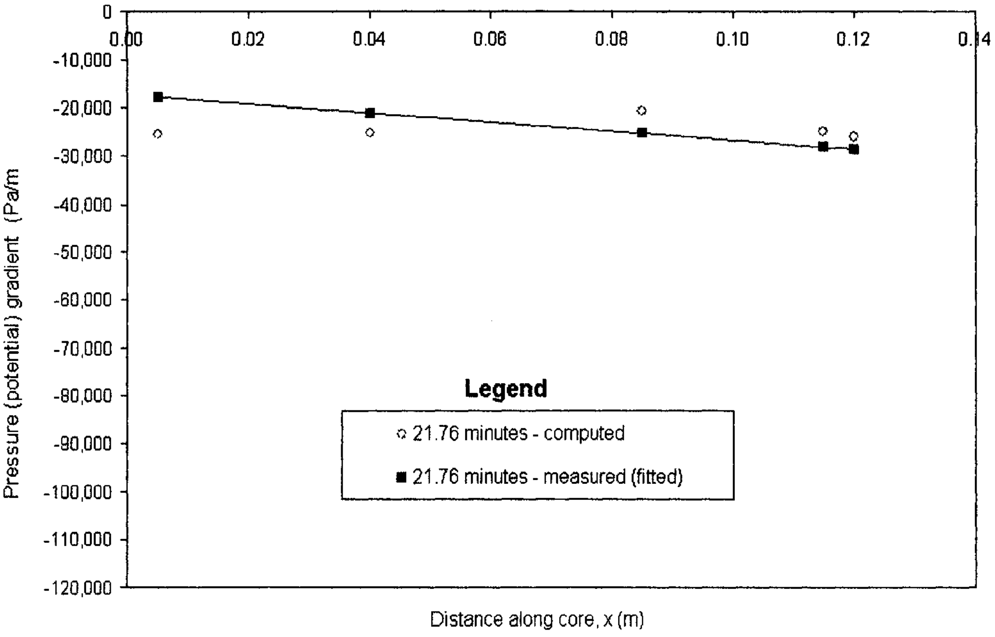


Figure 7.16e: Comparison of measured and computed nonwetting phase pressure curves along the core (Data Group B: Type II - USCO)

7.2.3.3. Data Group C: Bourbiaux and Kalaydjian (1990)

Data group C consists of some of the data sets presented by Bourbiaux and Kalaydjian (1990). The experimental data are made of USCO and USCC data and the data were acquired under the effect of capillarity and gravity (vertical flow in a vertical core). The data from these stable experimental tests were collected at laboratory conditions of 20^{0}C and a pressure close to atmospheric. The nonwetting phase is a paraffinic refined oil (Soltrol 130) and the wetting phase is water (brine). Interfacial tension between the two phases is 0.035 N/m. Details of core and fluid properties are shown in Table 7.3. Other data relevant to the discussion in this section are shown in Figures 7.17 to 7.24. These data were extracted from GVB-3 (one of the tests) data, presented by Bourbiaux and Kalaydjian. The X-ray absorption method was used for saturation measurements, and local water and oil pressures were transmitted through thin hydrophilic and hydrophobic semi-permeable membranes, respectively.

Table 7.3: Properties of Core and Fluids and Other Displacement Data for Data Group C

Core measurement length	0.29 m
Core cross-sectional area	0.0013 m^2
Core inclination, Θ	90.00 degree
Sand type (from Vosges region in France)	Fine, well-sorted & slightly Triassic
Average porosity	23.3%
Absolute permeability to wetting phase (brine)	120md ~ 115.44 x 10^{-15} m^2
Cocurrent effective permeability to non-wetting phase at S_{1i},	53.10 x 10^{-15} m^2
Countercurrent effec. perm. to non-wetting phase at S_{1i},	34.63 x 10^{-15} m^2
Cocurrent effective permeability to wetting phase at $S_{2r,}$	5.08 x 10^{-15} m^2
Countercurrent effec. perm. to wetting phase at $S_{2r,}$	3.53 x 10^{-15} m^2
Wetting phase (water/brine) viscosity at 20^{0}C	0.0012 Pa.s
Wetting phase (water/brine) density at 20^{0}C	1009 Kg/m^3
Non-wetting phase (paraffinic refined oil) viscosity at 20^{0}C	0.0015 Pa.s
Non-wetting phase (paraffinic refined oil) density at 20^{0}C	760.0 Kg/m^3
Initial wetting phase saturation	39.0%
Residual non-wetting phase saturation	37.6%
Flow type	USCO and USCC
Interfacial tension	0.035N/m
Capillary pressure-saturation data (curve)	See Figures 7.18
Cocurrent end point mobility ratio, M_{rco}	0.96422
Countercurrent end point mobility ratio, M_{rcc}	0.98017
Effective permeability/Relative permeability data (curve)	See Figures 7.17 and 7.24
Value of a for $R_{12} = 1 - a(1 - S)$ determination	0.00

The data presented in the figures were digitized from the original plots and others were processed from these digitized data. Bourbiaux and Kalaydjian (1990) presented cocurrent and countercurrent relative permeability data. The relative permeability data used were relative to the absolute permeability to brine. They inferred relative permeability data from simulations based on experimental production, saturation and pressure profiles of GVB-1, GVB-3 and GVB-4 tests, using the conventional multiphase Darcy law. The simulations were based on a one-dimensional, finite difference scheme. The cocurrent and countercurrent effective and relative permeability data presented in Figures 7.17 and 7.24 are based on the GVB-3 test (countercurrent experiment) and a simulated cocurrent equivalent of the GVB-3 test.

The original digitized relative permeability data (relative to absolute permeability to brine) were converted to effective permeability data (Figure 7.17) and then re-converted to relative permeability data (Figure 7.24). The reference (base) permeability is taken as the effective permeability to each phase at the residual saturation of the other phase (that is, at end point effective permeability values). The saturation values presented in all the figures are normalized saturation values computed from the original non-normalized saturation values presented by Bourbiaux and Kalaydjian (1990). Also, the fitted effective and relative permeability and capillary pressure data were determined by normalizing the saturation data presented by Bourbiaux and Kalaydjian (1990) and by fitting the effective permeability, relative permeability data and capillary pressure data against normalized saturation. Figures 7.17, 7.18 and 7.24 show the fitted effective permeability, relative permeability and the capillary pressure data, respectively. In all the figures (Figure 7.17 to 7.24) presented in this section, the solid lines indicate the fitted data, while the data symbols (triangle, circle and rectangle) indicate the digitized or processed digitized data points from the original plots.

It should be noted that the hydrodynamic effect was not included ($R_{12} = 1$) in the computation of the effective and relative permeability data. This is because Bourbiaux and Kalaydjian did not present details of pressure (potential) gradients in each phase, which are required to compute the magnitude of the hydrodynamic effects. Bentsen (1998a) had earlier showed that the non-inclusion of this effect does not introduce any significant error into the computation of relative or effective permeability for cocurrent flow but might contribute greater error to countercurrent flow.

Details of the experimental procedure and equipment descriptions can be found in Bourbiaux and Kalaydjian (1990). The analysis presented in the section below is based only on the analytical approach I (Section 7.2.1) because details of pressure and pressure gradient measurements in each phase were not presented by Bourbiaux and Kalaydjian.

7.2.3.3.1. Discussion of Data Group C Results (Approach I)

Figure 7.17 shows the original cocurrent and countercurrent effective permeability curves without any interfacial coupling correction applied. From the figure, the cocurrent effective permeability curves are much higher than the countercurrent effective permeability curves. Figure 7.19a shows the effective permeability when the capillary coupling correction was applied. From the results shown in Figure 7.19a, the cocurrent effective permeability curves, with the capillary coupling applied, did not overlap on the countercurrent curves but the difference between the curves narrowed slightly. The introduction of the capillary coupling term did not seem to lead to a better description of flow.

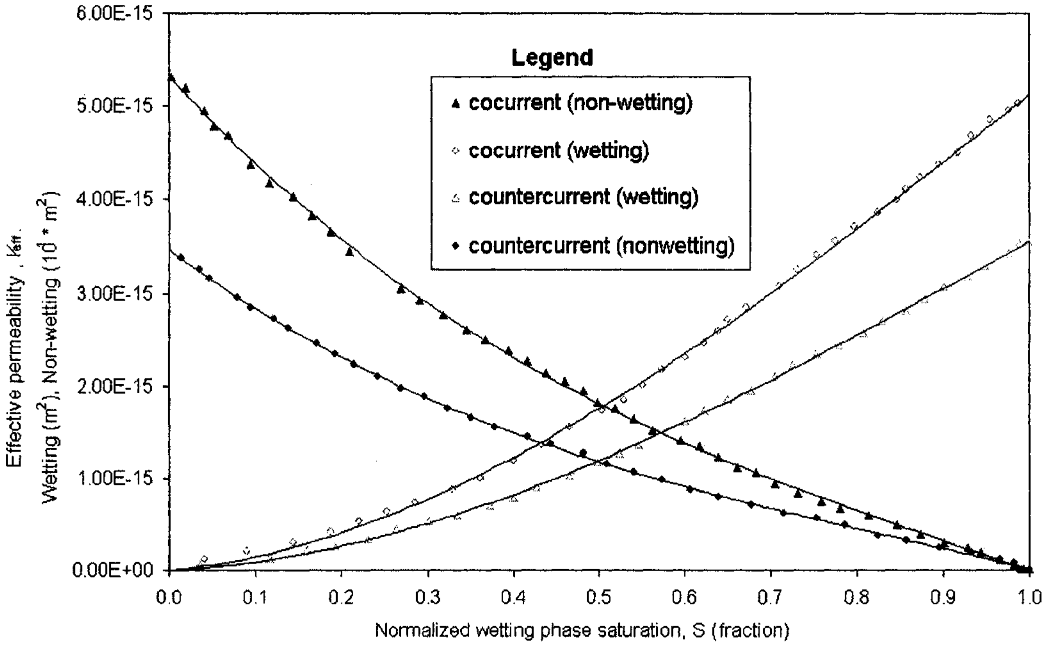


Figure 7.17: Effective permeability curves - Data Group C

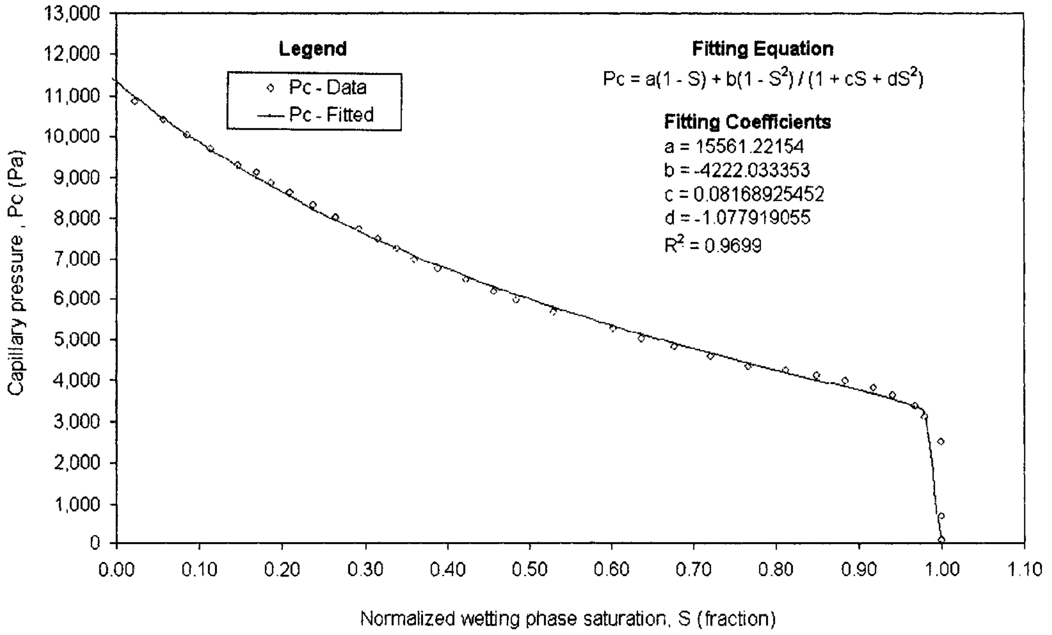


Figure 7.18: Capillary pressure-saturation curve - Data Group C

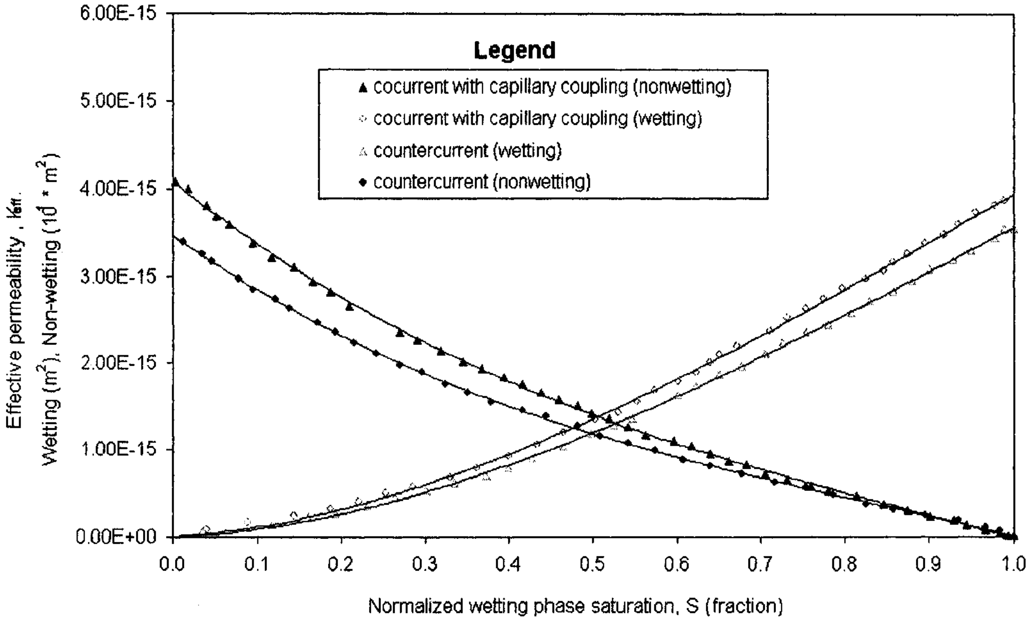


Figure 7.19a: Effective permeability curves with only capillary coupling correction applied to cocurrent flow - Data Group C

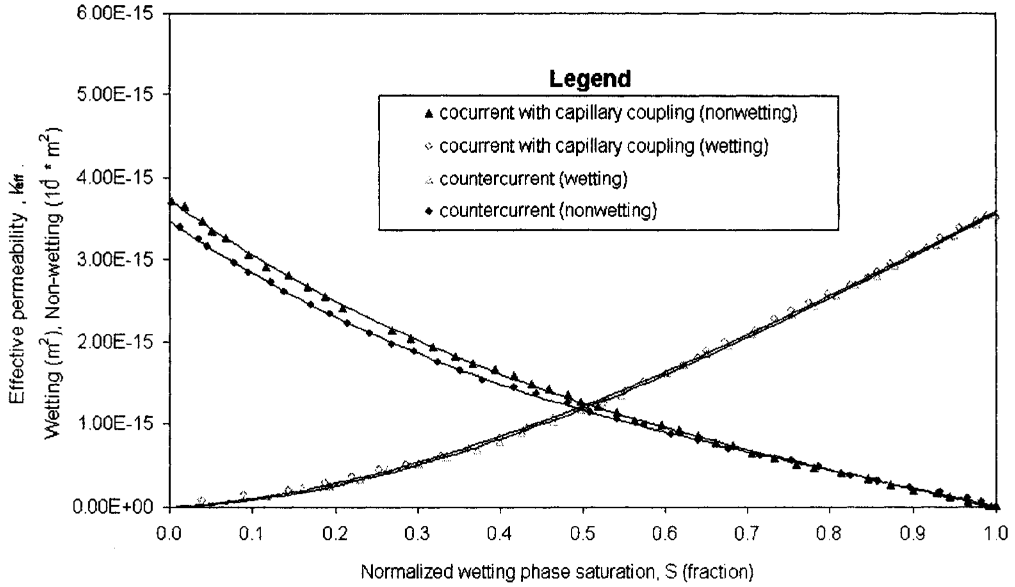


Figure 7.19b: Effective permeability curves with only assumed capillary coupling (1 - 0.3) correction applied to cocurrent flow - Data Group C

There are five possible explanations for this discrepancy. Firstly, the magnitude of the ratios of the end-point permeabilities of the wetting and nonwetting phases in the cocurrent and countercurrent flows differs although they are close. Secondly, the countercurrent relative permeability values reported by Bourbiaux and Kalaydjian (1990) were not obtained directly from Darcy's equation but through history matching by numerical simulation. Determining relative permeability through this approach always introduces significant error. Also, the cocurrent equivalent of the GB-3 cocurrent relative permeability data were obtained through history matching of the cocurrent data and not from actual experimental data. This approach is a simulation of simulation results, which leads to double errors or magnifications of errors in the determination of cocurrent relative permeability.

Thirdly, the value of porosity looks very small (23.3 %). This value might be correct because the sand system (porous medium) is consolidated, but most reported laboratory porosity values are in the 25 - 35 % range. If we assumed an average porosity value of 30% and use this as the basis of the capillary coupling correction term, the plot in Figure 7.19b results. This figure shows that at a more reasonable porosity value of 30%, the curves overlap for the wetting phase. The nonwetting phase curves show an improvement but did not completely overlap, especially at lower values of the wetting phase saturation.

Fourthly, the expression for capillary coupling obtained in Chapter 5 might be wrong. If that is so, then there might be a need to develop a more correct expression for the value of capillary coupling, especially in vertical flow. But it seems plausible that the capillary coupling expression is correct. No final conclusion can be made in this regard.

Lastly, neglecting hydrodynamic effects, especially in countercurrent flow might be another reason for the observed discrepancies. More experimental data in which the cocurrent and countercurrent experimental relative permeability data are determined directly, and hydrodynamic effects accounted for, are required to make any conclusion with regards to the validity of the capillary coupling expression, especially in vertical flow. It should be noted however that conducting cocurrent and countercurrent experiments with the same end point mobility ratio is extremely difficult. Future experiments should be geared towards acquisition of this type of data.

Again, applying the computed viscous coupling coefficients at each saturation value, the curves shown in Figure 7.20 were generated using Equations (7.5) and (7.6). Both the wetting and nonwetting effective permeability curves shifted slightly when the viscous coupling term was applied as shown. The values of c_1 and c_2 used in the computation of the viscous coupling term were taken to be equal to $2\phi^2$, that is, $c_1 = c_2 = c_{\max} = 2\phi^2$, the maximum value of c theoretically derived in Section 4.5.1.1. Using smaller values of $c_1 = c_2 = 0.001\phi^2$ leads to smaller differences between the cocurrent and countercurrent effective permeability values (Figure 7.21). So, again the results in Figure 7.20 and 7.21 show that the viscous coupling cannot be used to explain the larger differences in the values of cocurrent and countercurrent effective permeability.

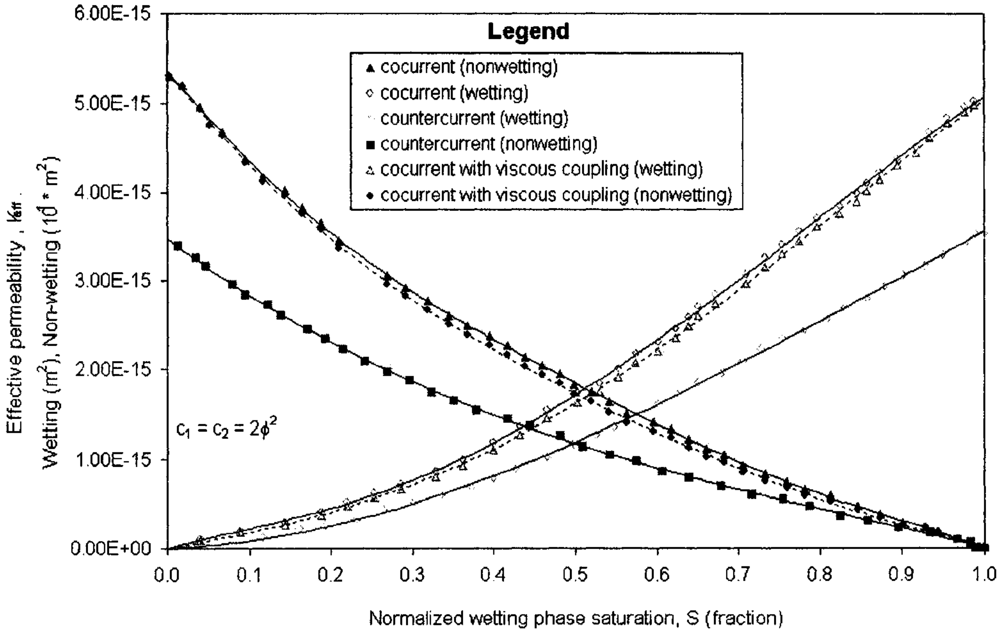


Figure 7.20: Effective permeability curves with only viscous (theoretical) coupling correction applied to cocurrent flow - Data Group C

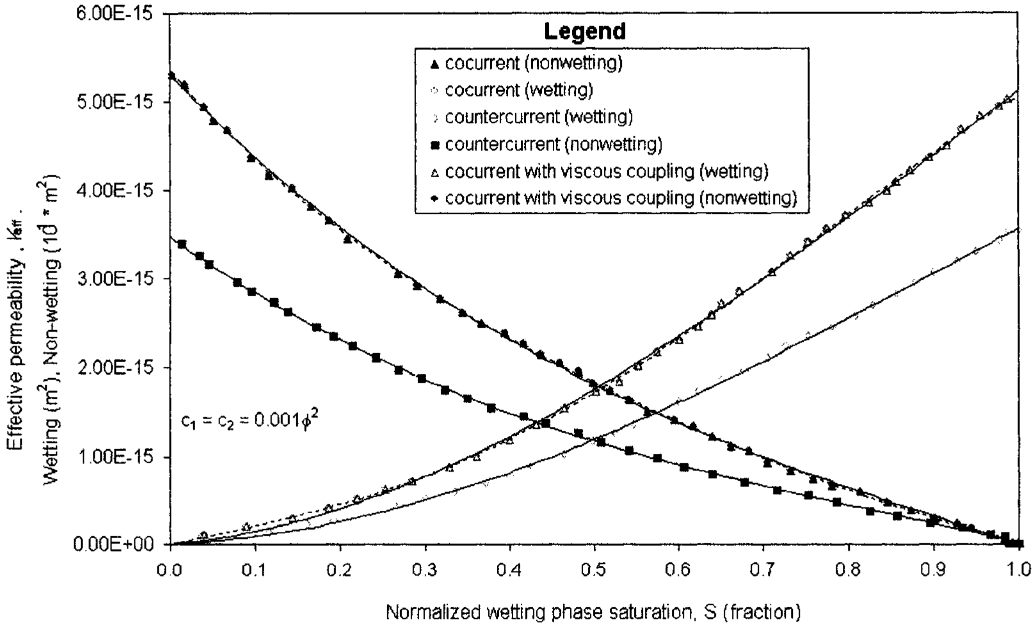


Figure 7.21: Effective permeability curves with only viscous (experimental) coupling correction applied to cocurrent flow - Data Group C

Figure 7.22 shows the generated curves when both the viscous and capillary coupling corrections were applied. The cocurrent and countercurrent curves did not overlap. This is due to the problems highlighted above as discussed for only the capillary coupling term. When the assumed 30% porosity value was used as the basis for capillary coupling, and viscous terms based on $c_1 = c_2 = 0.001\phi^2$, the cocurrent and countercurrent curves overlap for the wetting phase. The nonwetting phase curves show an improvement but did not completely overlap, especially at lower values of the wetting phase saturation (See Figure 7.23). As before, one cannot draw any conclusion on the validity of the porosity value and the capillary coupling term for the reasons highlighted previously.

Figure 7.24 shows the cocurrent and countercurrent effective permeability curves when they were normalized to the end point effective permeability values of each phase (that, is the relative permeability data). From the figures, both the cocurrent and countercurrent curves almost overlap. The slight difference between the two curves can be attributed to the presence of insignificant viscous coupling. Earlier investigators have noted this, and many have argued that what matters most in relative permeability data are the end point effective permeability values. The same phenomenon was observed with data group A in Section 7.2.3.1.1 but the viscous coupling effect is almost non-existent in this data set and it is not noticeable on the overlapped curves in Figure 7.1b.

Again, the end-point mobility ratio, M_r, is very important. The value of M_r depends on end point effective and relative permeability values to each phase. In the analysis in this section, the M_r values for cocurrent and countercurrent data are also approximately the same. Achieving the same end points required careful designs of the experiments.

7.2.4. Concluding Remarks on Analytical Verifications.

It can be concluded that the modified set of transport equations give a good description of flow and the interfacial coupling effects (capillary and viscous) that exist in horizontal, two-phase, porous media flow. Capillary coupling has a greater effect than viscous coupling in countercurrent flow, while both capillary and viscous couplings have insignificant effects in cocurrent flow. When transport equations are written in terms of effective permeability, the interfacial coupling effects should be included in the values determined using the expressions presented in this research, that is, Equations (7.1) to (7.6). It can also be inferred that either the cocurrent or the countercurrent relative permeability values can be determined experimentally and used in reservoir simulation.

If the conventional experimentally determined countercurrent effective permeability values are used in simulating a countercurrent flow recovery process, then the recovery prediction would be correct. Also, if the conventional experimentally determined countercurrent effective permeability values are used in simulating a cocurrent flow recovery process, then the countercurrent effective permeability values would have to be divided by the interfacial coupling parameter, $\alpha_i = \alpha_{ci} . \alpha_{vi}$, in order to obtain an accurate recovery prediction. On the other hand, the effective permeability values would have to be multiplied by the interfacial coupling parameter if the conventional experimentally determined cocurrent effective permeability values are used in the numerical simulation of a countercurrent flow. But no conclusion can be made with regards to capillary coupling in vertical flow. More experimental data would be required to make further deductions.

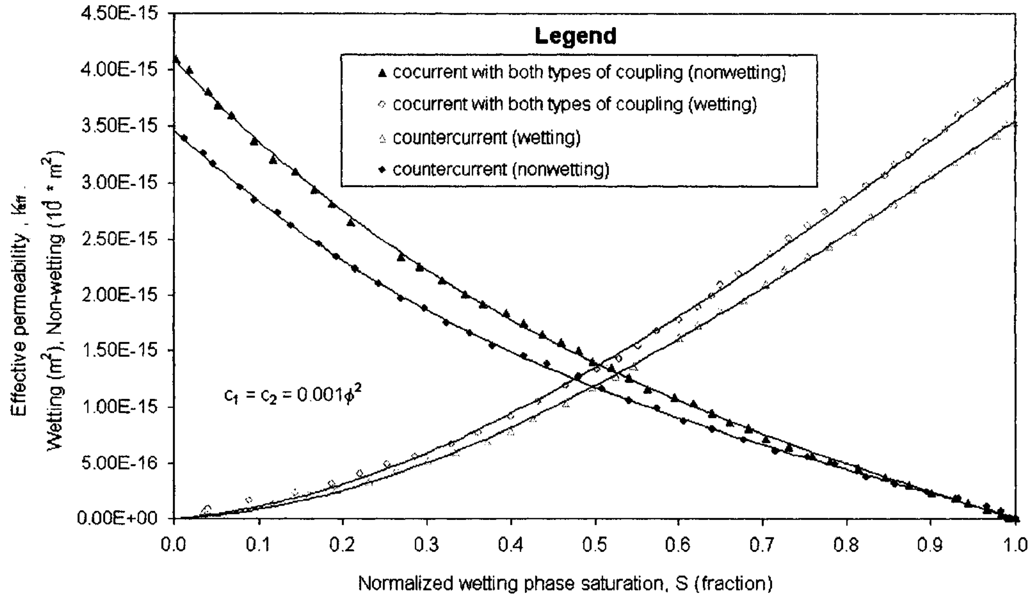


Figure 7.22: Effective permeability curves with both viscous (experimental) and capillary coupling corrections applied to cocurrent flow - Data Group C

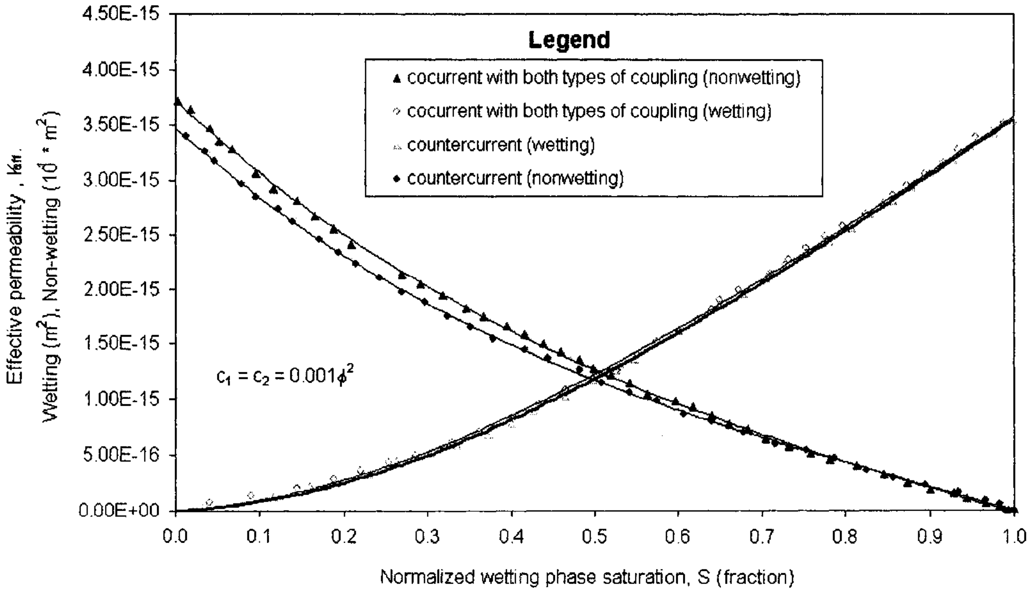


Figure 7.23: Effective permeability curves with both viscous (experimental) and with assumed capillary coupling (1 - 0.3) corrections applied to cocurrent flow - Data Group C

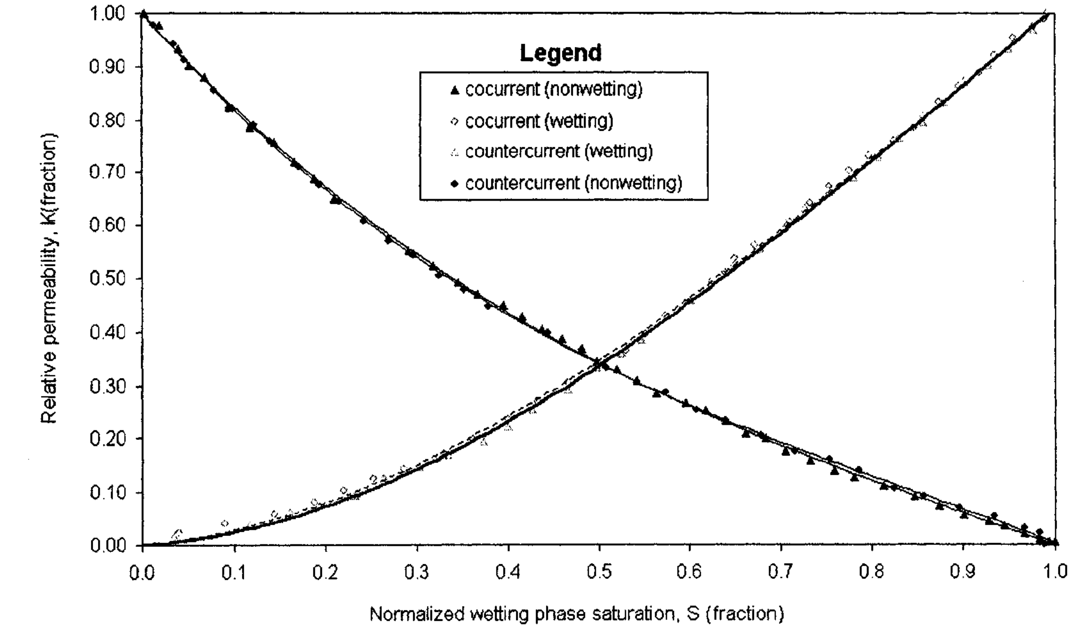


Figure 7.24: Relative permeability curves - Data Group C

7.3. Numerical Simulation Technique

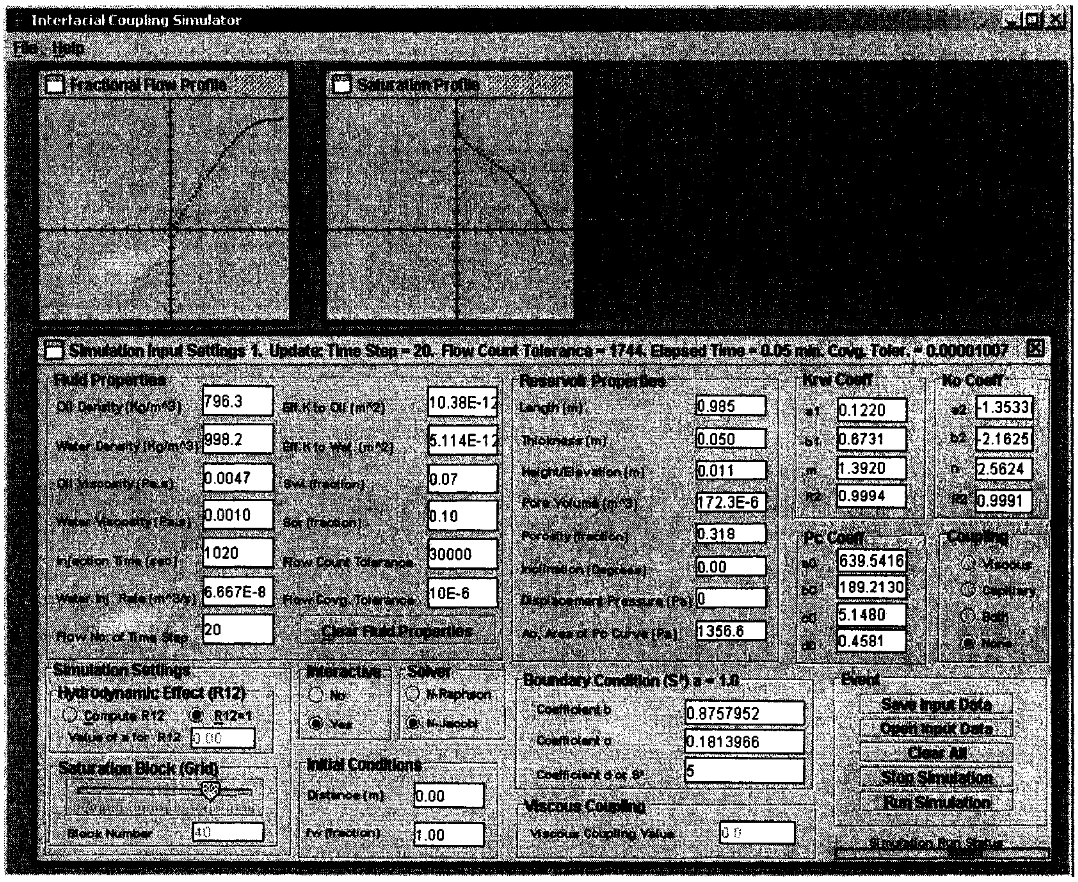
Another way to test or validate the modified set of transport equations is to compare the numerical solutions of the equations with actual laboratory experimental data. The development of a numerical simulator, with a graphical user interface, based on the equations is given in Chapter 5. That is, there is a need to compare the numerical simulation results from the simulator with the experimental data.

Apart from using the numerical simulator for sensitivity analysis to check the effects of neglecting any of the coupling effects, or the hydrodynamic effects that come into play, the numerical simulator can be used to validate the modified set of transport equations. This means that the simulator can be used to check if the transport equations, on which the simulator is built, model correctly the physics of flow as long as experiments with the same initial and boundary conditions as the numerical simulator runs are used.

To validate the equations, the relative permeability and the capillary pressure functions, obtained for a given set of experiments along with other relevant input parameters, are used as input into the simulator. See Figure 7.25 for the main input section of the simulator. After this, the simulator is run to obtain the production (fractional flow), saturation and potential (or pressure) distributions over a specified time interval. The output results from the simulator are stored in a text file in the default directory. This file can be exported to a spreadsheet for plotting. The simulator results are then compared with the actual fractional flow, potential (pressure) and saturation distributions obtained from the laboratory experiments. The equations can be said to model correctly the physics of multiphase flow if the experimental and the simulator fractional flow, potential and saturation distributions compare favourably, within the margin of error.

Before carrying out the actual verification tests, the effectiveness of the numerical simulator in reproducing cocurrent experimental data needs to be established through preliminary simulator check runs. Also, the limits of application of the numerical simulator, in terms of solution sensitivity to grid number and time steps, needs to be established. There is a need to test the limits of practicality through preliminary simulation check runs. The preliminary check runs and the actual verifications of the transport equations are presented below.

The first part of the preliminary check runs (Sections 7.3.1.1. to 7.3.1.6) covers fractional flow and saturation profile simulation, while the second part of the preliminary check runs (Sections 7.3.1.7) covers fractional flow, saturation and pressure gradient profiles. It is recommended that several combinations of grid number and time steps, for the given fluid-sand system being investigated, be used to ensure that the results are consistent for these combinations before being accepted as correct solutions. Some of the ways by which consistency can be achieved are explained in the sections below.



Simulation Input Settings 1. Update: Time Step = 20. Flow Count Tolerance = 1744. Elapsed Time = 0.05 min. Covg. Toler. = 0.00001007

Fluid Properties
Oil Density (Kg/m^3) 796.3
Water Density (Kg/m^3) 998.2
Oil Viscosity (Pa.s) 0.0047
Water Viscosity (Pa.s) 0.0010
Injection Time (sec) 1020
Water Inj. Rate (m^3/s) 6.667E-8
Flow No. of Time Step 20
Eff. K to Oil (m^2) 10.38E-12
Eff. K to Wat. (m^2) 5.114E-12
Swi (fraction) 0.07
Sor (fraction) 0.10
Flow Count Tolerance 30000
Flow Covg. Tolerance 10E-6

Reservoir Properties
Length (m) 0.985
Thickness (m) 0.050
Height/Elevation (m) 0.011
Pore Volume (m^3) 172.3E-6
Porosity (fraction) 0.318
Inclination (Degrees) 0.00
Displacement Pressure (Pa) 0
Ao, Area of Pc Curve (Pa) 1356.6

Krw Coeff
a1 0.1220
b1 0.6731
m 1.3920
R2 0.9994

Ko Coeff
a2 -1.3533
b2 -2.1625
n 2.5624
R2 0.9991

Pc Coeff
a0 639.5416
b0 189.2130
c0 5.1480
d0 0.4581

Coupling
○ Viscous
○ Capillary
○ Both
● None

Simulation Settings
Hydrodynamic Effect (R12)
○ Compute R12 ● R12=1
Value of a for R12 0.00

Interactive
○ No
● Yes

Solver
○ N-Raphson
● N-Jacobi

Saturation Block (Grid)
Block Number 40

Initial Conditions
Distance (m) 0.00
Fw (fraction) 1.00

Boundary Condition (S*) a = 1.0
Coefficient b 0.8757952
Coefficient c 0.1813966
Coefficient d or S* 5

Viscous Coupling
Viscous Coupling Value 0.0

Figure 7.25: Main input section (graphical user interface - GUI) of the simulator

7.3.1 Preliminary Numerical Simulation Runs (Check Runs)

For the preliminary testing, the experimental data presented in Section 7.2.3.1 were used. These experimental data were used also in the actual verification tests. Both the steady state and unsteady state experimental data presented in this section were used. The relative permeability and capillary pressure data were entered as fitted coefficients instead of as a table of values. This reduces computation time by eliminating table look-up operations during computation. Input data are shown in Tables 7.1 and 7.4. Table 7.1 contains data presented earlier in Sections 7.2.3.1 while Table 7.4 contains the fitting coefficients for the relative permeability, capillary pressure and the inlet saturation data for the core and fluids data in Section 7.2.3.1.

Table 7.4: Relative Permeability and Capillary Pressure Data Fitting Coefficients for Data Group A

Kr Coefficients		Pc Coefficients	Inlet Saturation Coefficients and Value
Wetting	Non-wetting	a = -46,639.5416	a = 1.0
a_1 = 0.1220	a_2 = -1.3533E+24	b = 66,189.2130	b = 0.8757952
b_1 = 0.6731	b_2 = -2.1625E+24	c = 5.148	c = 0.1813966
m = 1.3920	n = 2.5624	d = 0.4581	d = 5
R^2 = 0.9994	R^2 = 0.9991		Inlet Saturation, S* = 1.0

7.3.1.1. Comparison of Newton-Raphson and Newton-Jacobi Techniques

The numerical solutions of the fractional flow equation using the Newton-Raphson and Newton-Jacobi techniques, presented in Section 5.8.1, were compared to ascertain if they reproduce the same results. The results obtained with both techniques are compared with experimental data in Figure 7.26. The figure shows that the two techniques produced the same results and compared favourably with the experimental data. A grid number of 40 and a time step of 10,000 were used in both techniques. Arriving at the selected grid number and time steps was based on the analysis conducted and presented in Sections 7.3.1.3 and 7.3.1.4 below.

So, the use of a grid number of 40 and 10,000 time steps and above is guaranteed to produce convergent and consistent fractional flow results. The duration of the simulation was 20 seconds with the Newton-Raphson technique and 22 seconds using the Newton-Jacobi technique. This shows that both techniques converge at almost the same rate and either of the two techniques can be used. But the "N-Jacobi" option was arbitrarily selected in the "solver" panel in the GUI of Figure 7.25 in all of the numerical simulation runs presented below.

As the numerical solution procedure progresses, the solver option can be changed interactively (dynamically) between "N-Jacobi" and "N-Raphson" in the "interactive" panel of the GUI in Figure 7.25, if the interactive option is set to "yes". Also, if the "yes" button in the "interactive" panel of the GUI of Figure 7.25 is enabled, the fractional flow and the saturation profiles are plotted and viewed dynamically as the computation progresses.

7.3.1.2. Time of Running Simulation (Execution Time of Simulation)

For the selected time steps of 10,000 used in all the numerical results presented below, it takes approximately 22 to 37 seconds to carry out a single simulation run depending on the grid number. This also depends on the type of optimization carried out by the Java compiler (javac) itself based on the data structure being used under a given situation. This type of optimization is usually not visible to the user or programmer.

Also, for a given or fixed grid number, the execution time increases as the number of time steps increases. The execution time (for time steps of 10,000) might go as high as 62 seconds (1.03 minutes) or less if the interfacial option (both or either) is selected under the coupling option instead of the no coupling option and/or if the maximum grid number of 50 is employed. The saturation grid number can be in the range of 10 to 50. The "slider" in the GUI of Figure 7.25 in the "saturation block" panel can be used to vary the grid number in this range.

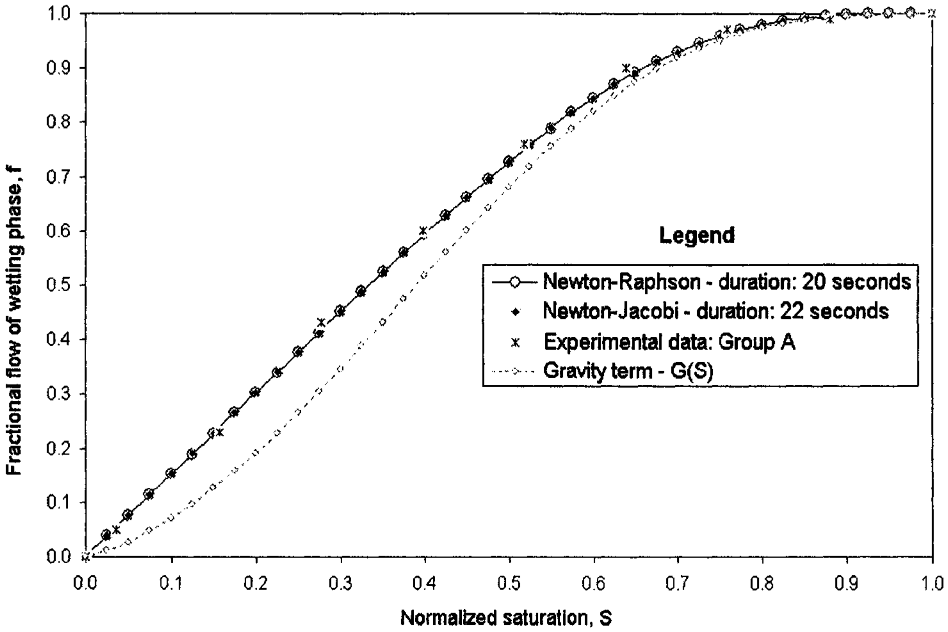


Figure 7.26 : Comparison of fractional flow profiles at 10,000 time steps for Newton-Raphson and Newton-Jacobi methods (Data Group A - cocurrent data)

7.3.1.3. Time Step Selection

The implicit and Crank-Nicholson schemes of Equations (5.54) and (5.71), respectively, are unconditionally stable but the time step for both equations is limited by accuracy requirements. That is, the time step is selected based on accuracy requirements not stability requirements. For the sand-fluid system under investigation, several runs were carried out to obtain solutions that meet accuracy requirements, that is, solutions close to the experimental results within the limits of experimental, truncation and round off errors.

The fractional flow profiles for grid numbers of 40 and 50 obtained from these runs are shown in Figures 7.27 and 7.28, respectively. From Figures 7.27 and 7.28, irrespective of the grid number (40 or 50), the fractional flow profiles are the same as long as the time steps are kept in the interval, 20 to 14,000. On the other hand, a time step in the range 10,000 to 14,000 was found to be most suitable for predicting saturation profiles that compare favorably with experimental saturation profiles, that is, profiles that meet the accuracy requirements (See Figure 7.29 for results obtained for a grid number of 40). This narrows further the above intervals for fractional flow. So, it is recommended that any grid number and time steps in the range 10,000 to 14,000 be used in all simulation runs. This recommendation was used as a guide in the analysis in Section 7.3.2.

If the simulator is used for carrying out numerical studies on a different fluid-sand system, different lower and upper bounds of the values of the time step might be obtained, which are different from 10,000 to 14,000, respectively, due to the accuracy requirements stated above. In such situations, it is recommended that several combinations of time steps and grid number be investigated to ensure that consistent results are obtained in order to establish the correct lower and upper bound values.

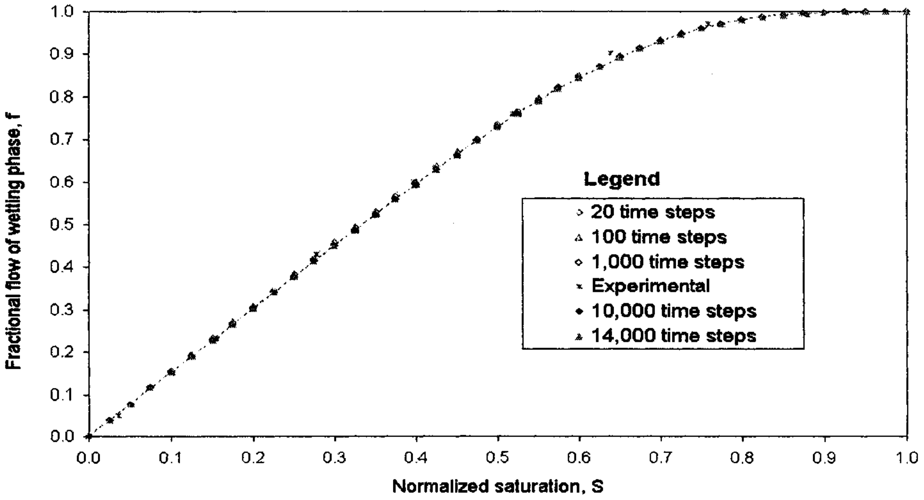


Figure 7.27 : Comparison of fractional flow profiles at various time steps using the Newton-Jacobi solution method at a grid number of 40
(Data Group A - cocurrent data)

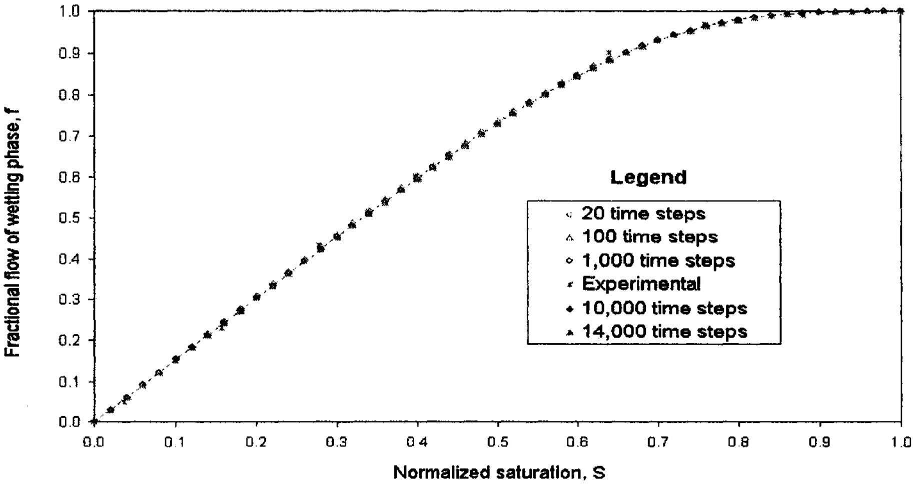


Figure 7.28 : Comparison of fractional flow profiles at various time steps using the Newton-Jacobi solution method at a grid number of 50
(Data Group A - cocurrent data)

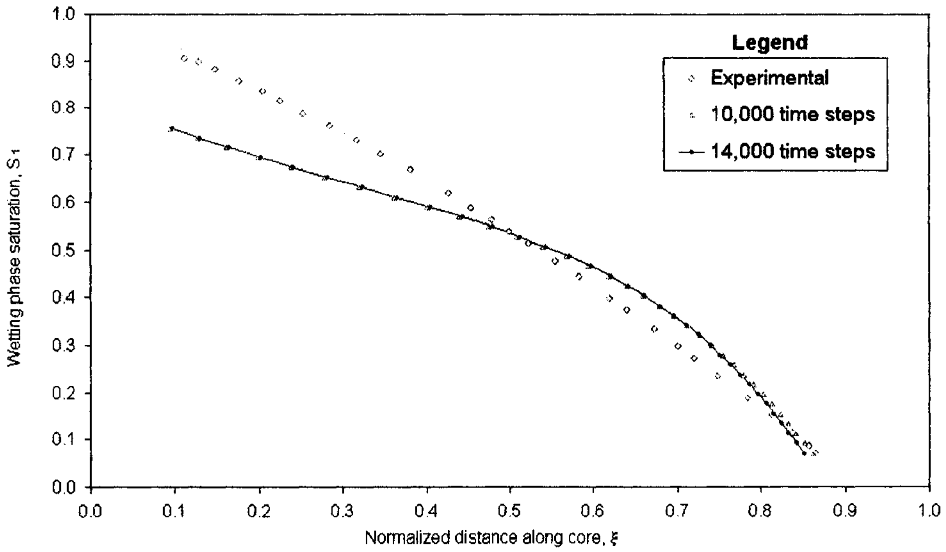


Figure 7.29 : Comparison of experimental saturation profiles with numerical saturation profiles at 10,000 and 14,000 time steps at a grid number of 40 (Data Group A - cocurrent data)

7.3.1.4. Saturation Grid Number Sensitivity

For the results presented in Section 7.3.1.7 and 7.3.2.1 to 7.3.2.4, a time step of 10,000 was used because it falls in the range 10,000 to 14,000. The solution sensitivity to saturation grid number was investigated using grid numbers of 40 and 50. The results are shown in Figures 7.27 and 7.28. In the plots shown in these figures, it can be seen that in the range 10,000 to 14,000 time steps, under the same input conditions and data at the indicted grid numbers, the output fractional flows are basically the same, irrespective of the grid number. This shows that as long as the correct time step is selected, subject to accuracy requirements, the results are guaranteed to be the same irrespective of the grid number.

Using a grid size beyond 10 placed higher burdens on the computer memory and time of computation. The higher computation time at a higher grid number was due partly to the use of a Java pure object-oriented approach and partly due to computation overheads of Java arrays indexing. The software makes extensive use of arrays to handle, manipulate and store data. Every array indexing requires bound checking that contributes to computation overhead. Using a higher grid number results in using a higher number of array elements and consequently a higher computation time, although the execution time is relatively small (of the order of a few seconds or minutes) as discussed in Section 7.3.1.2

7.3.1.5. Inlet Saturation Assumption

The algorithm developed in Chapter 5 allows the possibility of starting the initially guessed value of the normalized inlet saturation at any small value, say 0.05, and then iterating continuously until a correct value of the inlet saturation is obtained at the correct converged (flow convergence) values and the material balance condition. But, in reality, one knows the inlet saturation, or an approximate value of the inlet saturation, so in the coding of the algorithms, one can input directly this value. This is because experiments are conducted in such a way that the inlet saturation at a given time is recorded and known.

It was found that the solution of the fractional flow equation might converge to the wrong solution if an iterative solution technique is employed. For example, an initially assumed iterated value of 0.4, which is supposed to be 0.64, might converge before the iteration step reaches 0.64 and give a good material balance, resulting in the wrong fractional flow profile. Hence, there is no need to carry out a material balance computation in the iterative solution step.

There are two options for specifying the inlet saturation. Firstly, if one has a set of data that can be fitted to Equation (5.51), the coefficients used in fitting Equation (5.51) are supplied as input parameters in the text boxes of the "boundary conditions" panel of the GUI in Figure 7.25. These coefficients are in turn used to compute the inlet saturation with Equation (5.51). Therefore, the use of Equation (5.51) renders the material balance approach redundant. Secondly, or alternatively, if one knows the inlet saturation at a given time, the coefficients b and c in the "boundary conditions" panel are specified as 0.0 and 0.0 in the textboxes for coefficient b and c, while the actual inlet saturation value is specified in the textbox for coefficient d. The simulator automatically knows this and it uses the value as the inlet saturation value.

7.3.1.6. Convergence Tolerance (Decimal Approximation)

Round off error that accumulates during numerical computation depends on the number of decimal places used in the computation. To minimize this, the difference between successive Newton-Raphson and Newton-Jacobi iterative approximations (that is, the residual) of the fractional flow data point for solution convergence was set at 10^{-6}. This value can be changed to a smaller or higher value in the "flow convergence tolerance" text box of the GUI in Figure 7.25. The value, 10^{-6}, was found to be most suitable for data group A check runs. This value was based on the optimal selection of grid number, computer memory requirements and computation time.

7.3.1.7. Preliminary Comparison of Experimental and Numerical Fractional Flow, Saturation and Pressure Gradient Profiles

With the input data in Tables 7.1 and 7.4, a simulation run was carried out for a preliminary comparison of experimental and numerical results based on an optimum selection of grid number and time steps as detailed above. The run is applicable to cocurrent flow and interfacial coupling and hydrodynamic effects were not included. A grid number of 40, time steps of 10,000 and the Newton-Jacobi solution technique were employed, based on the results presented in Section 7.3.1. The results are shown in Figures 7.30, 7.31, and 7.32.

Comparisons of fractional flow experimental and numerical results show excellent agreement as shown in Figure 7.30. From Figure 7.31, although the numerical solution of the saturation profile did not match the experimental results exactly, the agreement between the two is reasonable and this is accepted within the margin of error. As seen in Figure 7.31, the numerical solution spreads out (oscillates about the experimental data); that is, the differences between the two balance out at the top and bottom regions and the numerical results can be fairly accepted as having the same area under the curve as the experimental results. Also the material balance gives a value of 0.0108 (about 2.27% of the normalized time), which is good within the margin of experimental, numerical and curve fitting errors. The material balance error, which should be zero for a perfect run, is the absolute value of the difference between the normalized time and the area under the saturation-distance curve.

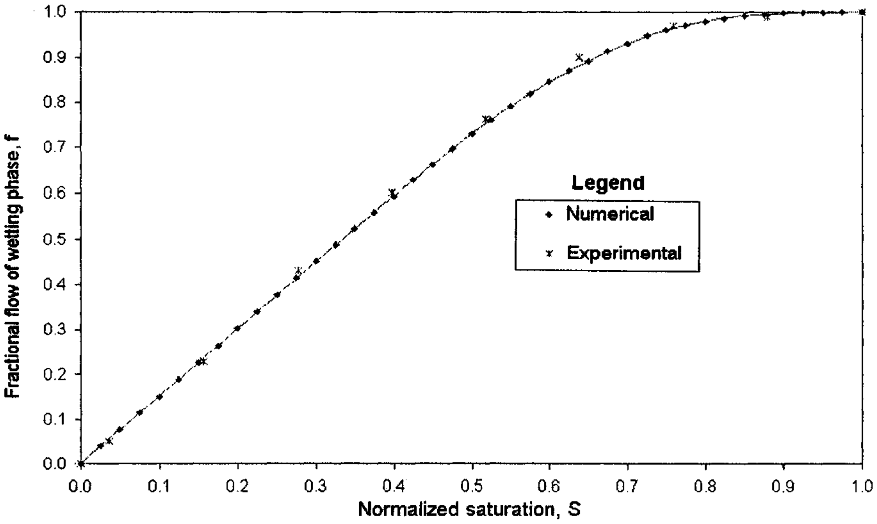


Figure 7.30 : Comparison of experimental and numerical fractional flow profiles without interfacial coupling and hydrodynamic effects (Data Group A - cocurrent data)

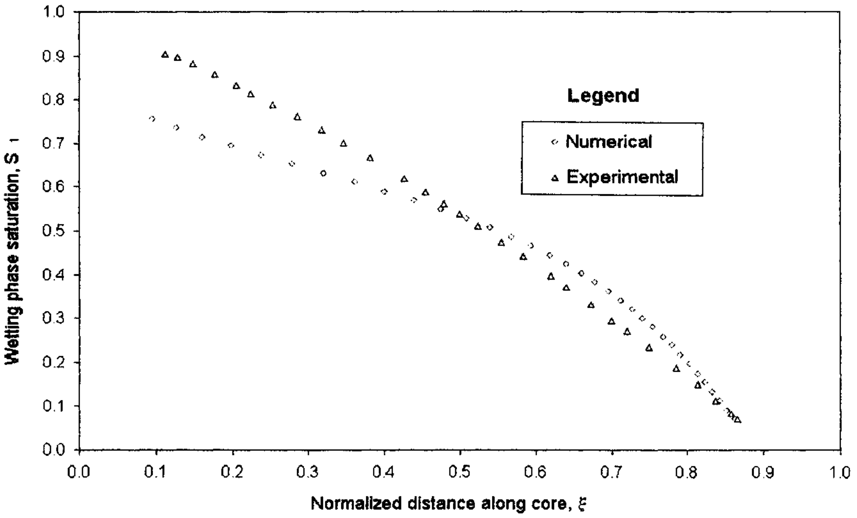


Figure 7.31 : Comparison of experimental and numerical saturation profiles without interfacial coupling and hydrodynamic effects (Data Group A - cocurrent data)

Comparisons of the experimental and numerical pressure gradients in the wetting and non-wetting phases are shown in Figure 7.32. Experimental and numerical pressure gradient data were compared up to 0.75m (equivalent) of the core, the point the last pressure sensor was located to obtain the original experimental data presented by Sarma and Bentsen (1989a). The two sets of data show good agreement, although some deviations are noticeable, especially towards the inlet end and outlet ends of the core. Such deviations were due partly to experimental error and partly to deviation in the computation of saturation, which were transferred to the computation of the pressure gradient profiles. Such was the case, because the pressure gradient computations make use of the saturation profiles' results.

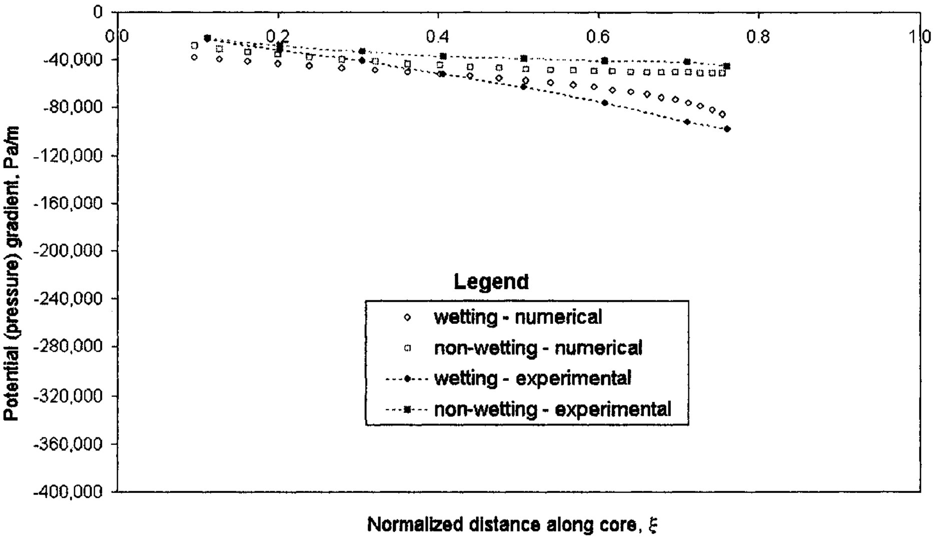


Figure 7.32 : Comparison of experimental and numerical pressure gradient profiles without interfacial coupling and hydrodynamic effects
(Data Group A - cocurrent data)

7.3.2 Actual Validation with Numerical Simulation Runs with Data Group A

With the input data in Tables 7.1 and 7.4, several simulation runs were carried out for the actual validation. Interfacial coupling and hydrodynamic effects were both included and neglected to check their impacts. In all the numerical results presented below, a grid number of 40, time steps of 10,000 and the Newton-Jacobi solution technique were employed, based on the results presented in Section 7.3.1. For the results presented in Sections 7.3.2.1 to 7.3.2.4, an unsteady state time level of 1020 seconds was used, while for the results in Section 7.3.2.5, unsteady state time levels of 500, 300, 150, 100 and 50 seconds were used.

The concept of fractional flow breaks down for countercurrent flow; hence, the numerical simulator cannot handle the countercurrent flow situation and countercurrent flow simulation runs were not performed. This is because in countercurrent flow, the wetting phase and non-wetting phase flow rates and potential gradients are opposite in sign and there exists some saturation values for which the total flow rates of both phases is zero and the fractional flow values become undefined.

7.3.2.1. Neglecting the Hydrodynamic Effect in Cocurrent Flow with No Coupling

Figures 7.33, 7.34 and 7.35, respectively, show a comparison of the fractional flow profiles, saturation profiles and pressure gradient distributions when hydrodynamic effects were included and neglected in the simulation runs with the no interfacial coupling option selected. The cocurrent end point effective permeability values were used for the hydrodynamic and non-hydrodynamic effects' runs. From the figures, it can be seen that there is little or no difference between the profiles whether hydrodynamic effects are included or not. The fractional flow and the saturation profiles match exactly while the saturation and pressure gradient profiles show negligible differences (average of less than 0.93%). This shows that neglecting hydrodynamic effects in cocurrent flow did not introduce any significant error into the predicted production (fractional flow), saturation and pressure profiles. These results reinforced earlier observations obtained using the analytical verification techniques presented in Section 7.2.3 as well as observations by previous researchers (Bentsen (1998a) and Ayub (2000)).

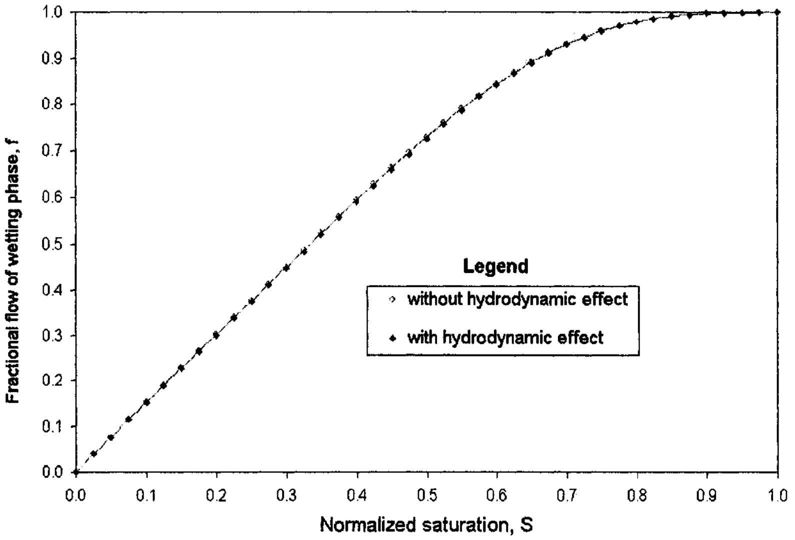


Figure 7.33 : Comparison of fractional flow profiles with and without hydrodynamic effects (Data Group A - cocurrent data - no interfacial coupling effect included)

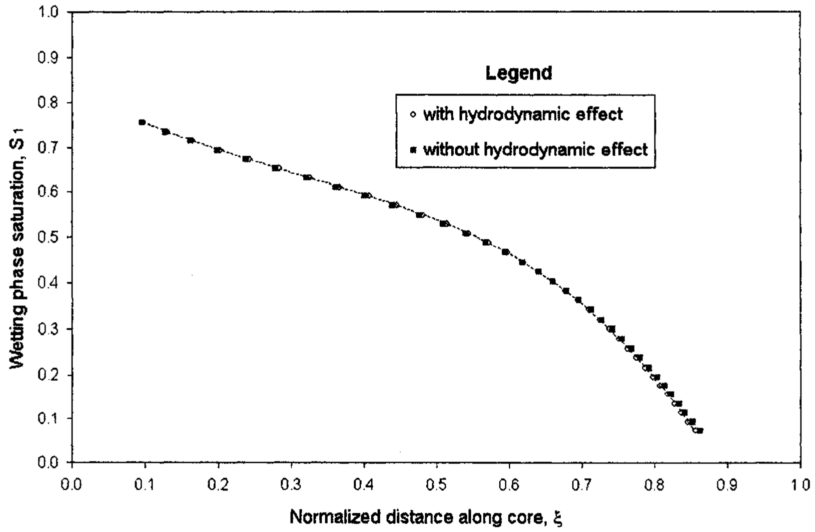


Figure 7.34 : Comparison of saturation profiles with and without hydrodynamic effects (Data Group A - cocurrent data - no interfacial coupling effect included)

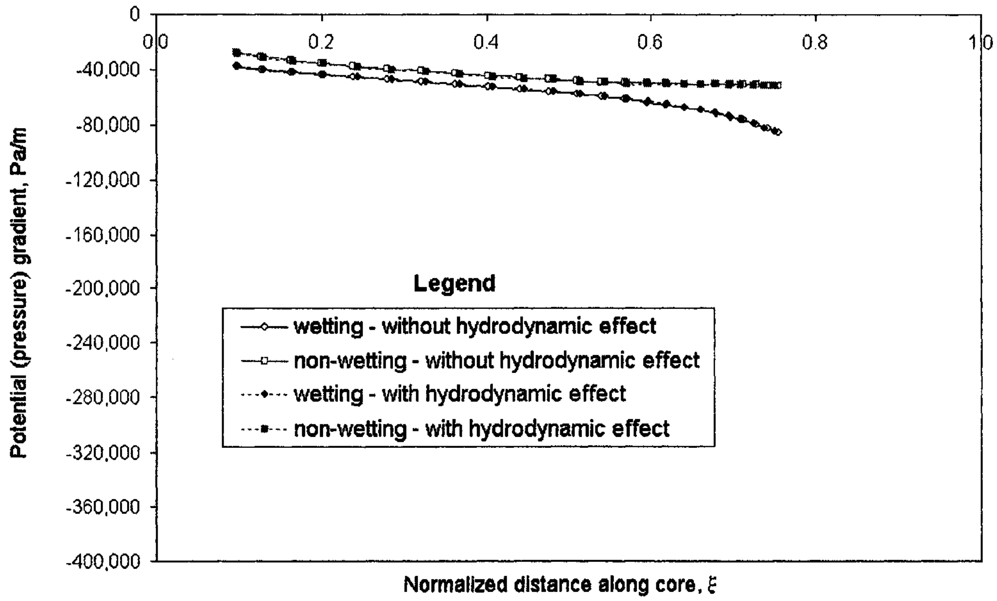


Figure 7.35 : Comparison of of pressure gradient profiles with and without hydrodynamic effects (Data Group A - cocurrent data - no interfacial coupling effect included)

7.3.2.2. Neglecting Either or Both Types of Coupling with and without Hydrodynamic Effects

Here, the same input data as in Section 7.3.2.1 were used but with "viscous", "capillary", or "both" selected in the "coupling" option panel of the GUI in Figure 7.25. The hydrodynamic effects were not included and the parameter that controls the amount of viscous coupling was taken as $c_1 = c_2 \approx 0.001\phi^2$ (See section 7.2.3.1.1).

The results when only viscous coupling was selected are shown in Figures 7.36, 7.37 and 7.38 and the results when only capillary coupling was selected are shown in Figures 7.39, 7.40 and 7.41. When the "both" couplings option was selected, the predicted profiles are shown in Figures 7.42, 7.43 and 7.44.

From the figures, there is little difference between the predicted profiles for capillary coupling and both types of coupling. Also, there is no difference or undetectable differences between the predicted profiles for only viscous coupling. The results show that neglecting either or both of the interfacial coupling effects does not contribute any significant effect to the predicted profiles in cocurrent, horizontal, two-phase, porous media flow.

Also, when hydrodynamic effects were included, the predicted results are the same as those shown in Figures 7.36 to 7.44. This again confirms that the hydrodynamic effect can be neglected.

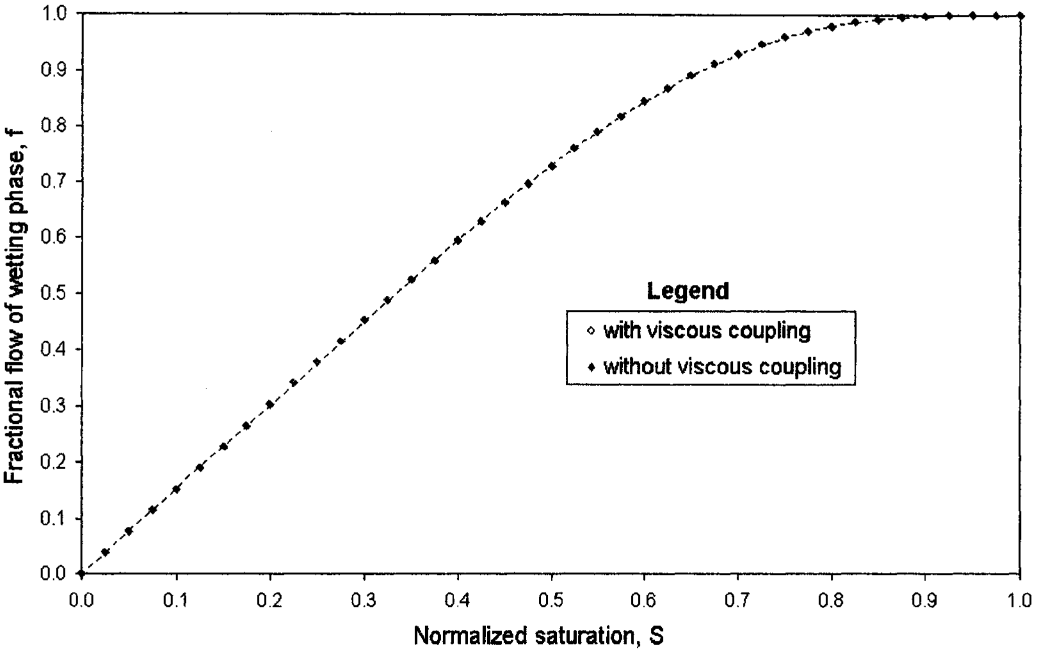


Figure 7.36 : Comparison of fractional flow profiles with and without viscous coupling (Data Group A)

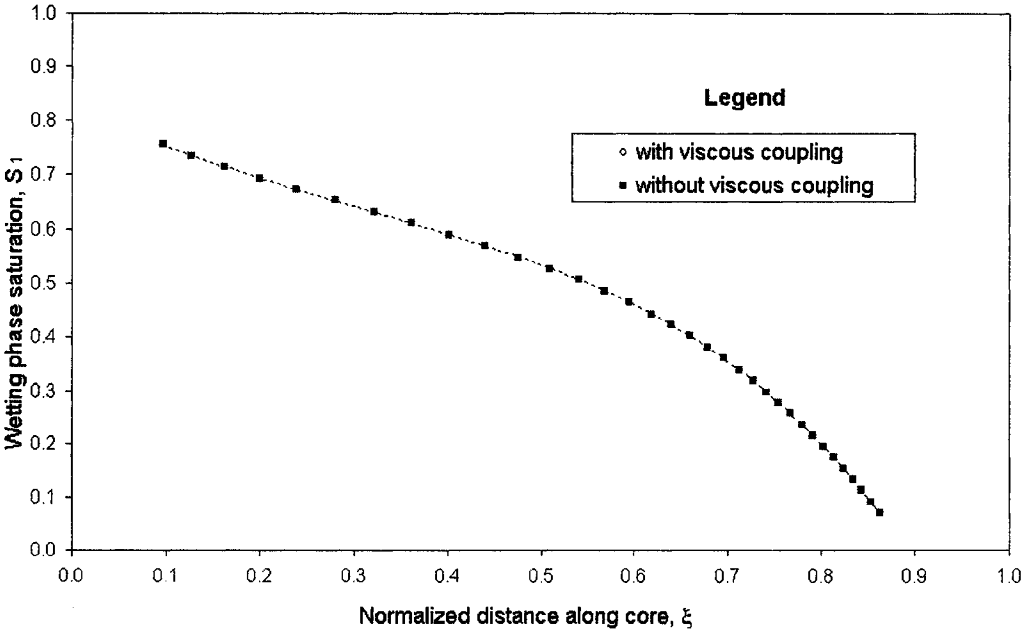


Figure 7.37 : Comparison of saturation profiles with and without viscous coupling (Data Group A)

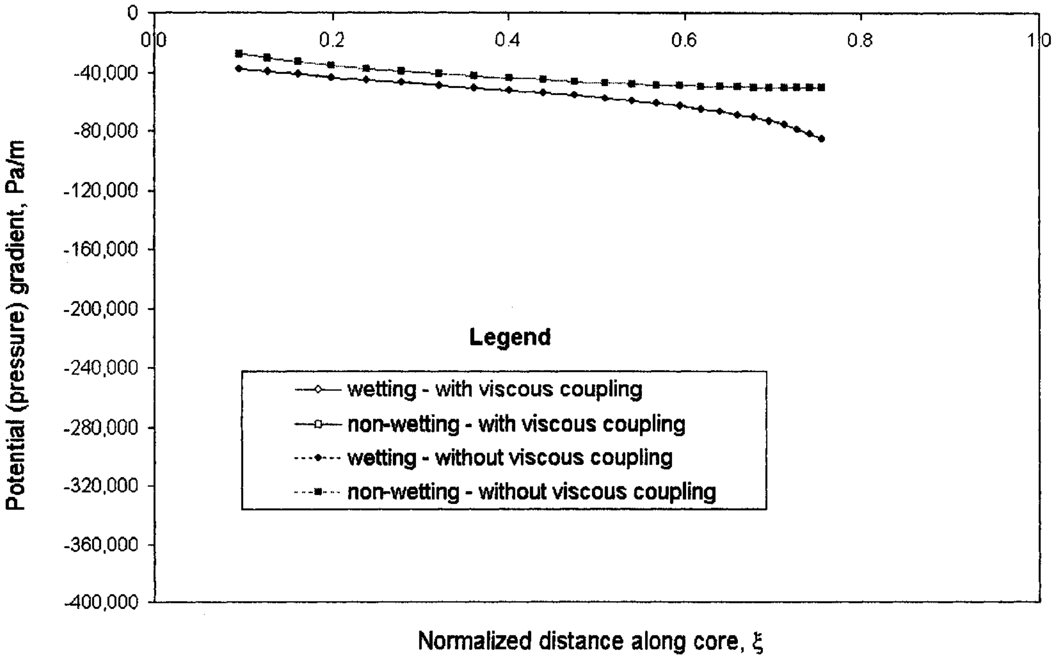


Figure 7.38 : Comparison of pressure gradient with and without viscous coupling (Data Group A)

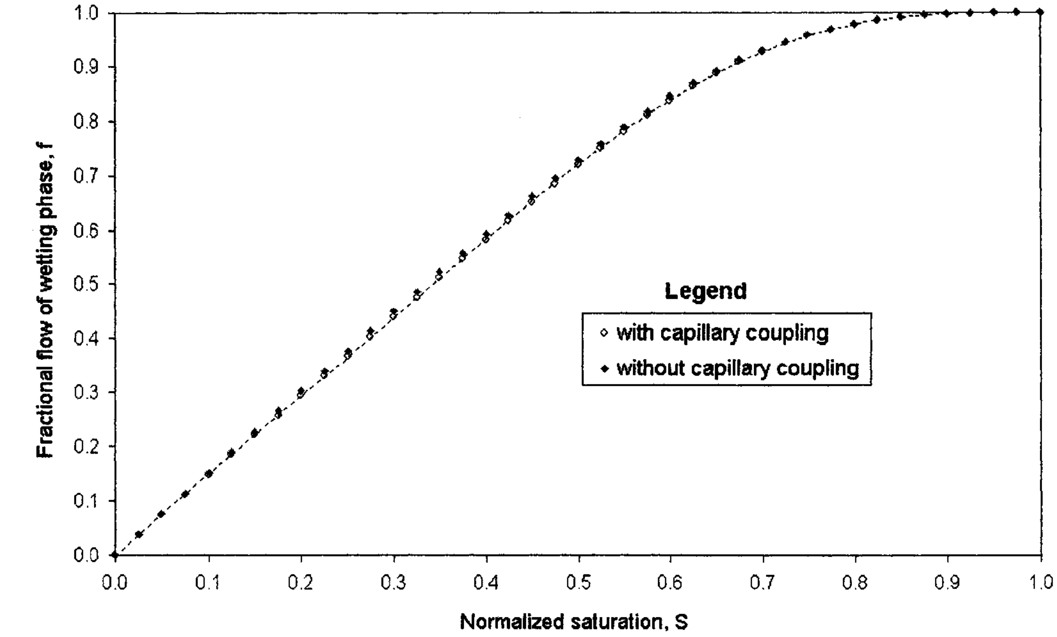


Figure 7.39 : Comparison of fractional flow profiles with and without capillary coupling (Data Group A)

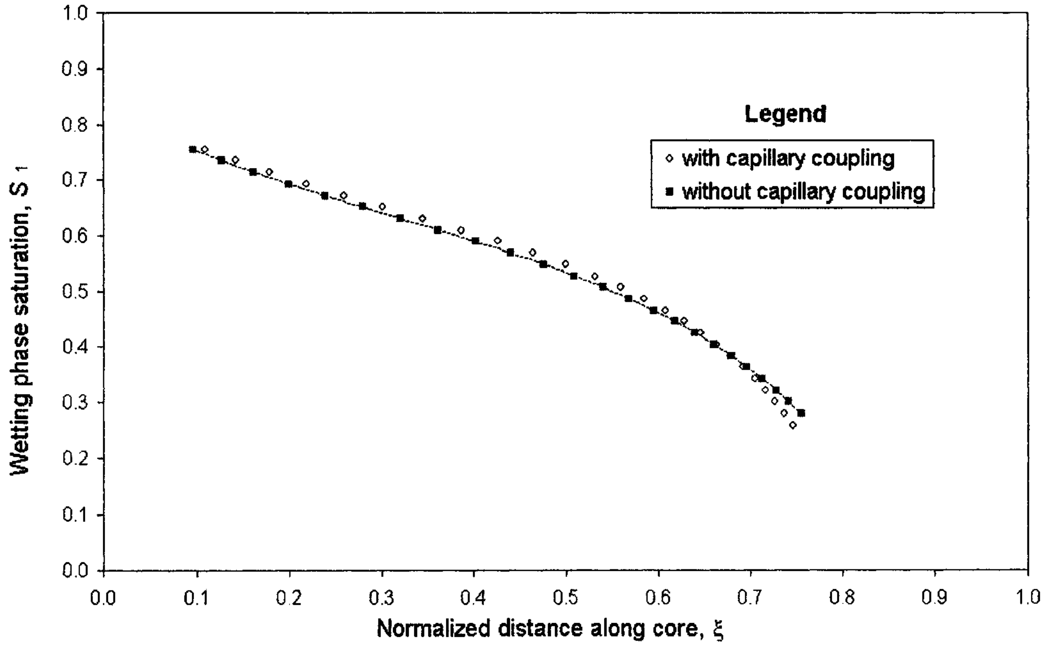


Figure 7.40 : Comparison of saturation profiles with and without capillary coupling (Data Group A)

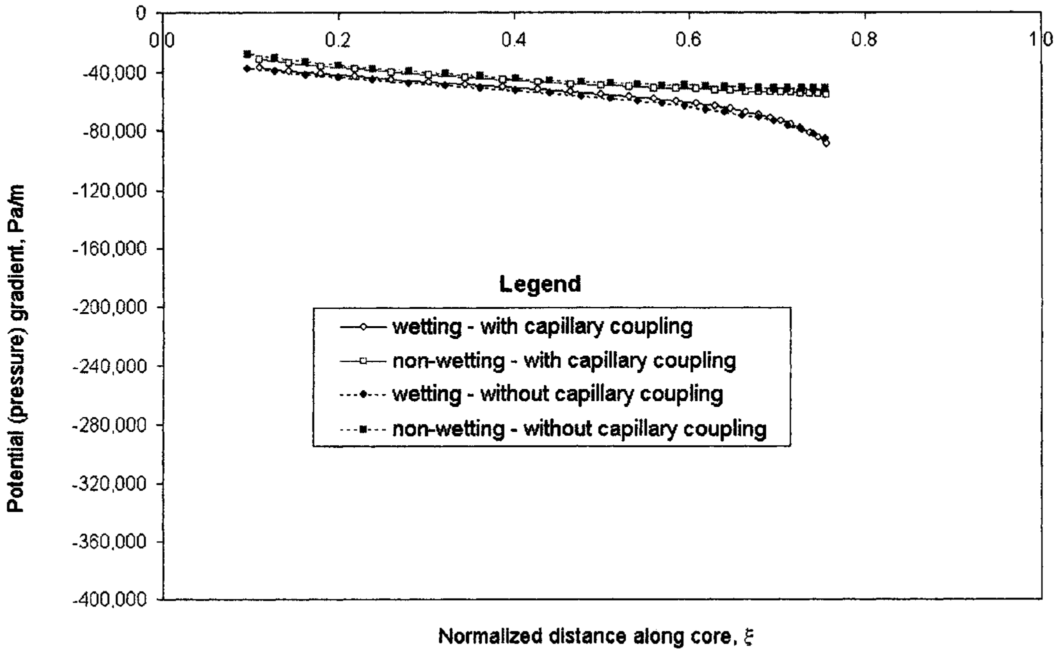


Figure 7.41 : Comparison of pressure gradient with and without capillary coupling (Data Group A)

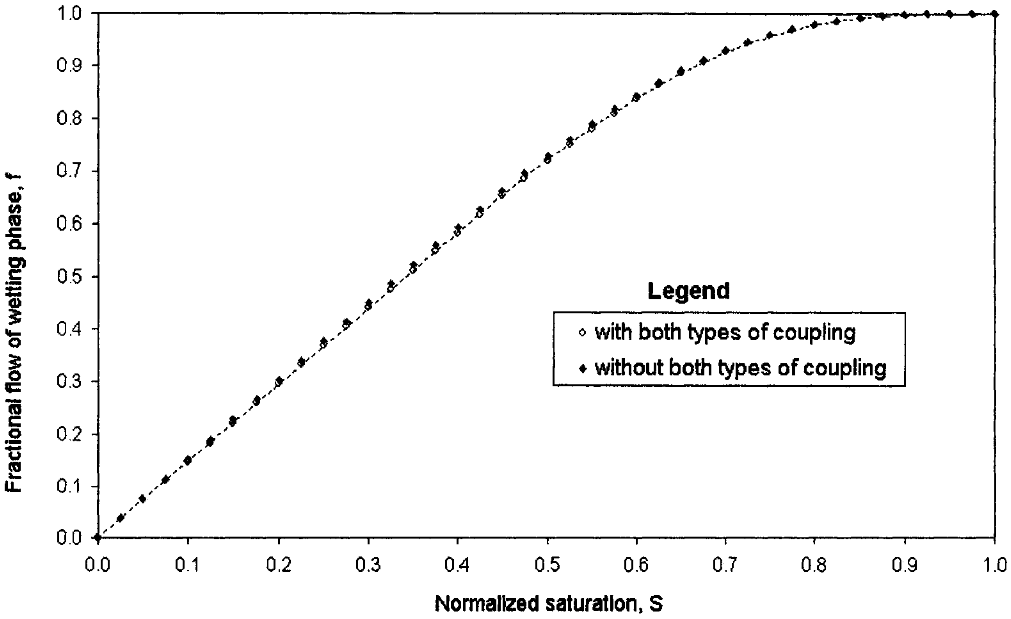


Figure 7.42 : Comparison of fractional flow profiles with and without both types of coupling (Data Group A)

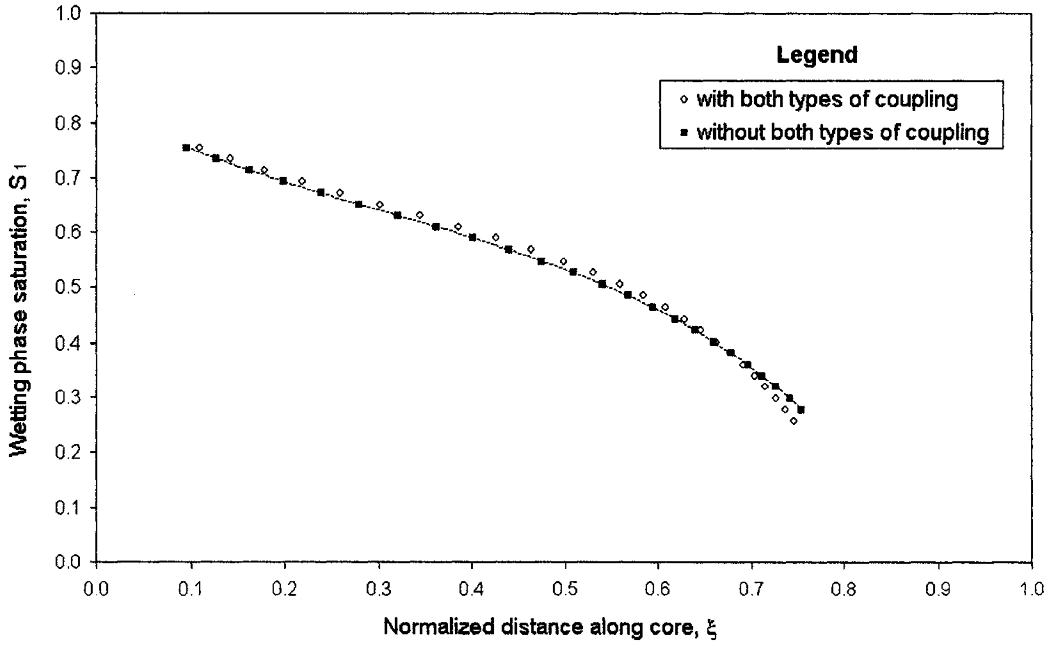


Figure 7.43 : Comparison of saturation profiles with and without both types of coupling (Data Group A)

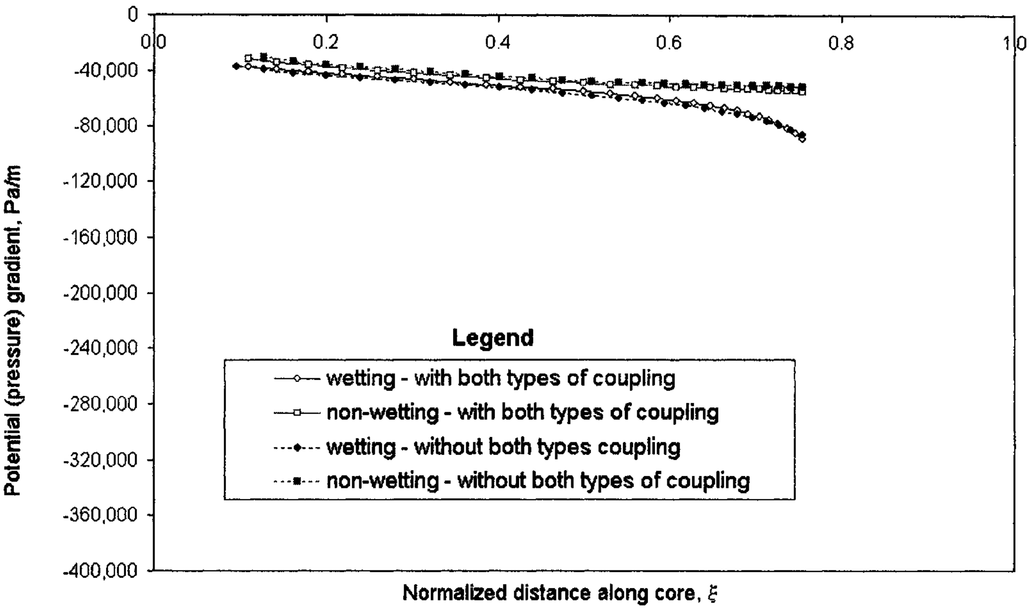


Figure 7.44 : Comparison of pressure gradient with and without both types of coupling (Data Group A)

7.3.2.3. Sensitivity Analysis of Viscous Coupling

Figures 7.45, 7.46 and 7.47 show the results of the sensitivity analysis when the viscous coupling parameter was varied with no hydrodynamic effects. In the sensitivity analysis runs, the input data were the same as those of Section 7.3.2.1 except that the viscous option was selected in the "coupling" option panel instead of "none". Also, the parameter that controls the amount of viscous coupling was varied. The expression for the parameter is given as $X\phi^2$, where the constant X was found theoretically to have a maximum value of 2 (See Equations 4.57 and 4.58). By using various values of X, which are 20, 10, 5, 2, and 0.001, the viscous coupling parameter was varied in the simulation runs.

From the results generated (Figure 7.45), the fractional flow is under predicted when high values of X were used. The fractional flow gradually rises as the value reduces from 20 to about 2 and then 0.001, at which values, the predicted fractional profiles are the same as the predicted profile when the interfacial-coupling effect was not included. Also, the absolute values of the saturation profiles (Figure 7.46) and the pressure gradient profiles (Figure 7.47) are over predicted towards the inlet end and middle of the core and under predicted close to the outlet end of the core at higher values of X. As the value of X reduces from 20 to about 2 and then 0.001 the predicted saturation and pressure gradient profiles are close to or almost the same as the predicted profiles when the interfacial-coupling effect was not included.

Theoretically, the value of X was found to have a maximum value of 2 (Section 4.5.1.1, Equations 4.57 and 4.58) and as a result of this, it seems improbable that X can rise above 2. But in the present situation, raising the value of X as high as 20, helps one to quantify what the effects of viscous coupling would be if X rises as high as 20. Hence, it can be inferred that viscous coupling has no significant effect on the predicted profiles and can be neglected in cocurrent; horizontal, two-phase, porous media flow if the parameter that controls the amount of viscous coupling is equal to or below the maximum theoretical value, which is 2.

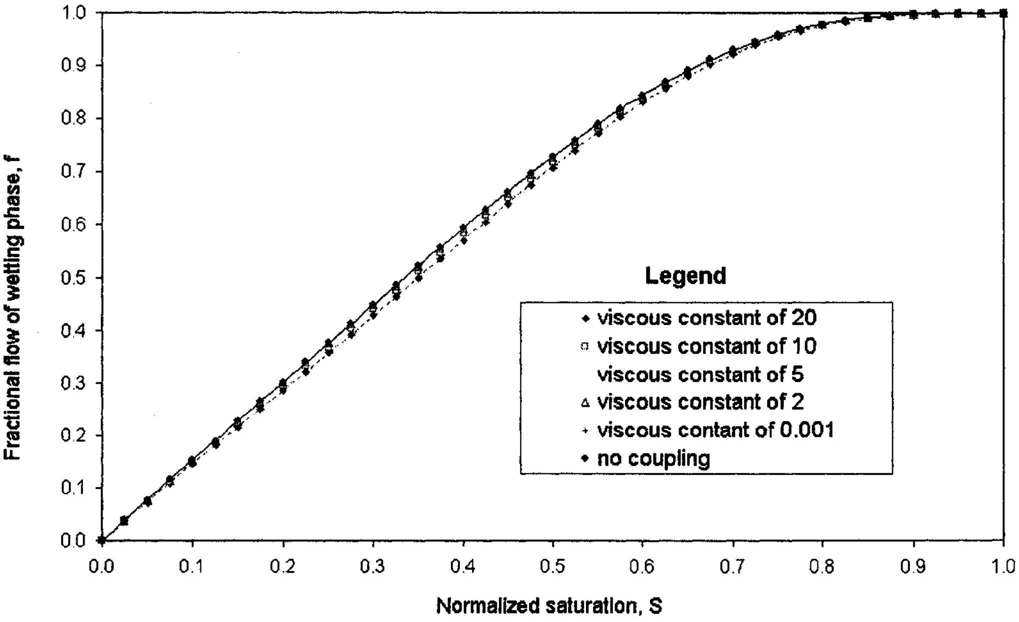


Figure 7.45 : Effect of viscous coupling on fractional flow profiles (Data Group A)

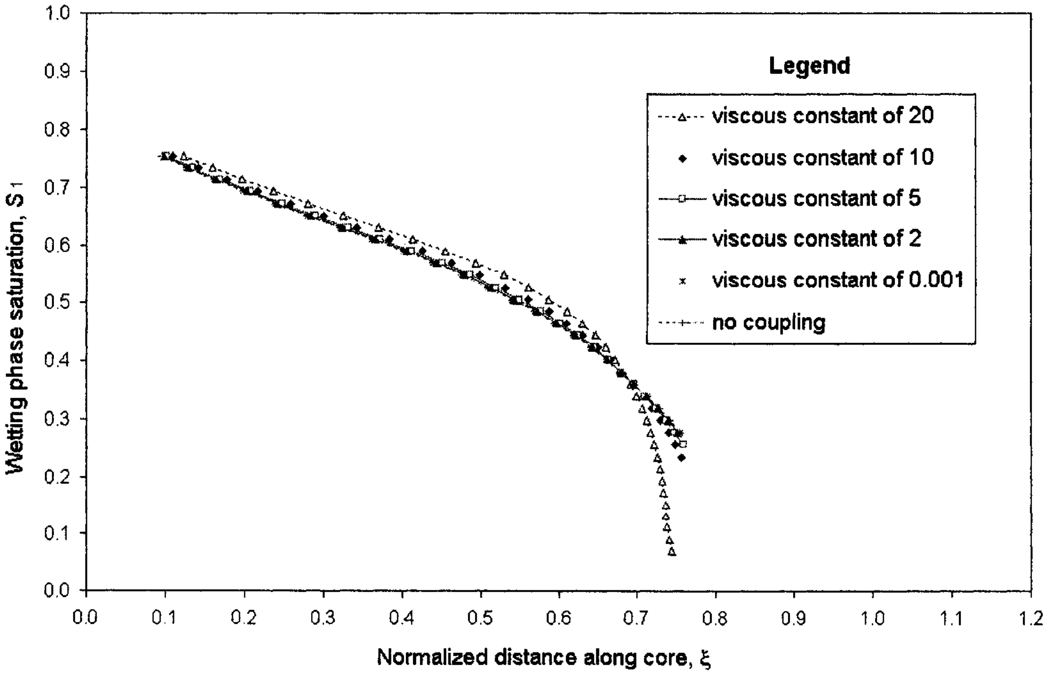


Figure 7.46 : Effect of viscous coupling on saturation profiles (Data Group A)

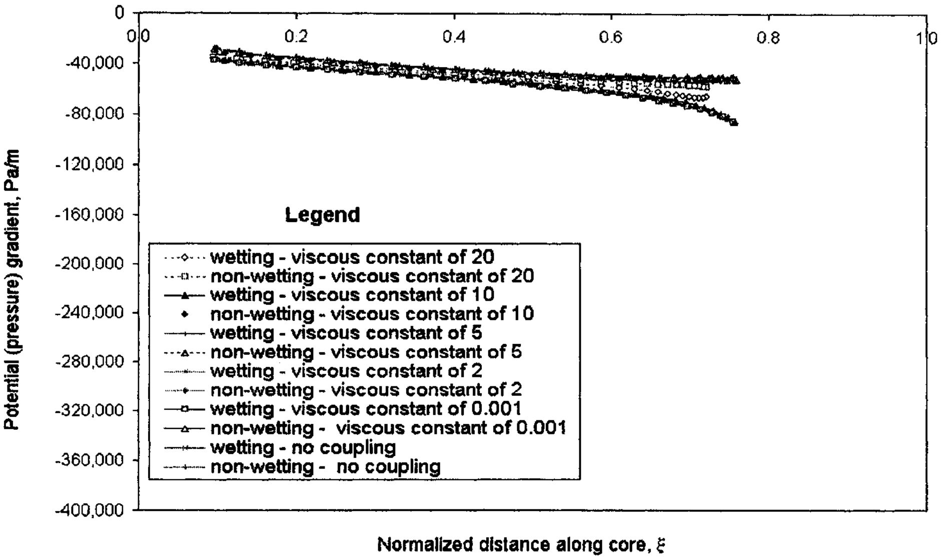


Figure 7.47 : Effect of viscous coupling on pressure gradient profiles (Data Group A)

7.3.2.4. Sensitivity Analysis of Both Types of Coupling

Here, the same input data as in Section 7.3.2.3 were used, but with the "both" coupling option selected. The capillary coupling term is constant, within the saturation range, at $1-\phi$, while the viscous coupling parameter was varied by using the various values of the parameter that control the amount of viscous coupling as in Section 7.3.2.3. Figures 7.48, 7.49 and 7.50 show the results generated when the simulation runs were carried out. The results are essentially similar to the results generated in section 7.3.2.3 except that the profiles deviate slightly from those of Figures 7.45, 7.46 and 7.47. The deviations are very small.

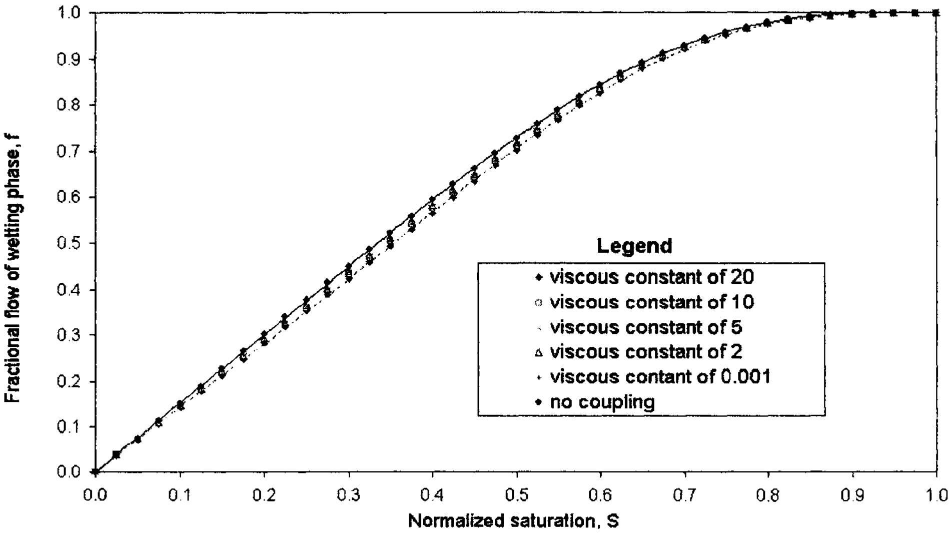


Figure 7.48 :Effect of both types of coupling on fractional flow profiles(Data Group A)

Again, at the theoretical value of X, which is 2, and below, the fractional flow, saturation and potential gradient profiles are quite close and almost the same. This shows the effects of capillary coupling and viscous coupling in cocurrent, horizontal, two-phase, porous media flow, are insignificant, although the introduction of the capillary coupling along with viscous coupling gives an insignificant effect as compared to when only the viscous coupling was introduced into the simulation runs.

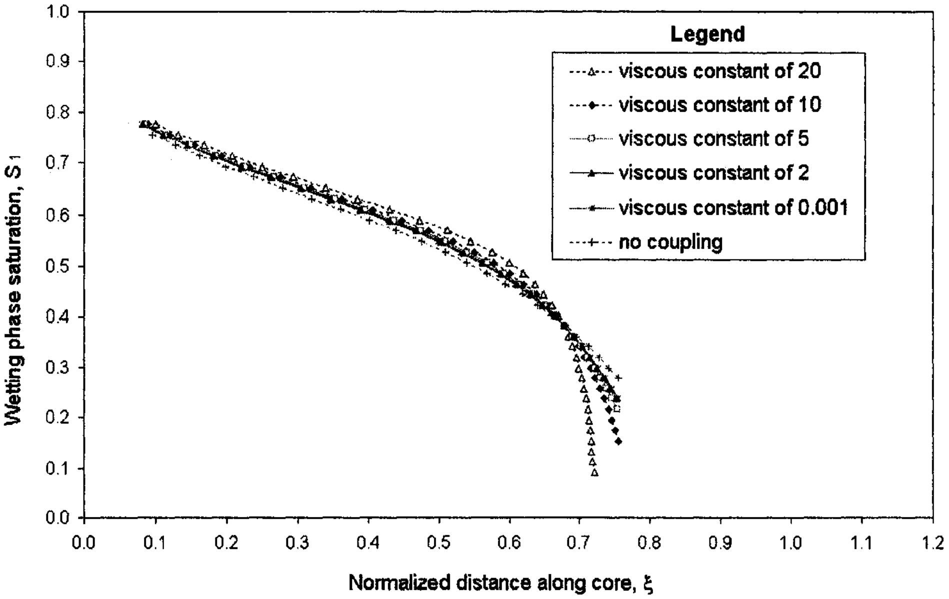


Figure 7.49 : Effect of both types of coupling on saturation profiles (Data Group A)

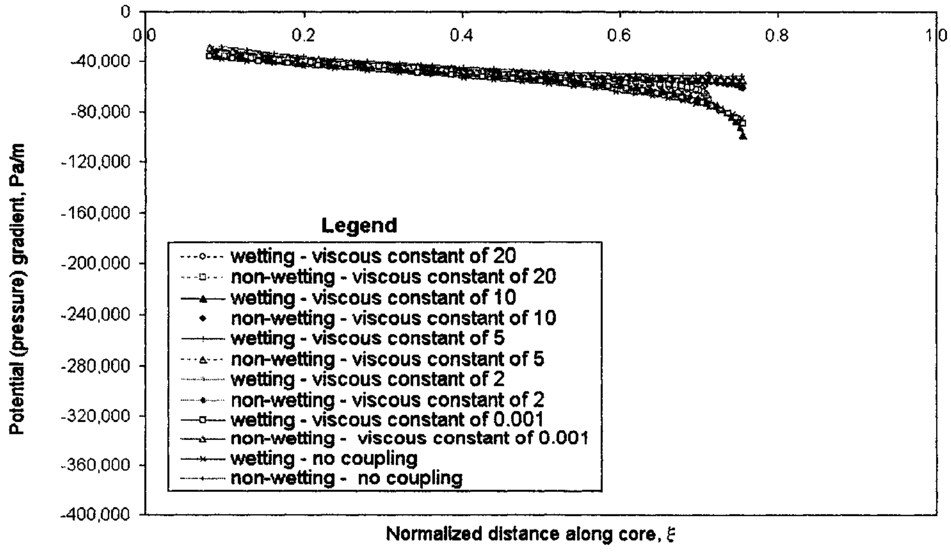


Figure 7.50 : Effect of both types of coupling on pressure gradient profiles (Data Group A)

7.3.2.5. Saturation Profiles Over time

Having carried out the actual validation with numerical simulation runs, additional runs were carried out to check how saturation profiles evolved with time. In all the numerical results given in 7.3.1, the simulation time was 1020 seconds and computing the inlet saturation based on the coefficients in Equation 5.51 and Table 7.4 always results into a saturation value in 1.0 (which is the saturation discretization domain). Certain behaviours with regards to the rise of the inlet saturation with time were observed when a lower simulation time, which gives a discretization domain of less than 1.0, were used.

Firstly, using Equation 5.51 with a lesser time, say 400 seconds, which gives a value of saturation, on which saturations were discretized, gives a piston-like displacement except when the computed value is close to 1.0, say 0.89 and above. The behaviour was thought to be due to the fact that the numerical algorithm could not effectively deal with situations where inlet saturations that are based on Equation 5.51 are used as the discretization domain.

Secondly, it was observed that using a saturation value of 1.0 as the discretization domain is acceptable and the solution gives acceptable saturation profiles, based on the developed numerical algorithm. The saturation profiles based on this approach actually helps to show the inlet saturation that corresponds to a given time interval. In this regard, the numerical simulator is an indicator of the inlet saturation under a given set of displacement conditions at a specified time.

Hence, the additional runs carried out to check how the saturation profiles evolved over time were based on the above observation. That is the discretized saturation domain was set between the initial wetting phase saturation and the residual wetting phase saturation, which corresponds to 0 and 1.0 normalized wetting phase saturation, respectively. Results from these additional runs are shown in Figure 7.52. The input data are the same as the data in Tables 7.1 and 7.2, without hydrodynamic effects included under cocurrent flow conditions, except that some flow conditions were changed to make the displacement stabilized.

The changes are as follows: flow rate was reduced to 2.667 x 10^{-8} m^3/s, the times of simulation are 8000, 6000, 3000, 500 and 50, and the length of the core was increased to 2.0 m. In addition, the relative permeability data were modified by changing some of the coefficients used to fit the relative permeability data in Table 7.4. The coefficients are m and n, and they were changed to 2.0. The modified relative permeability data and the original relative permeability data are plotted in Figure 7.51. With the modified data, the capillary number, Nc, becomes 0. 07. A grid-number of 40 was used and a time step of 10,000 was selected.

The material balance error values for simulation times of 8000, 6000, 3000, 500 and 50 are, respectively, 0.0124, 0.0096, 0.0055, 0.0012, and 0.0002. When expressed as percentages of the normalized times, the values give 1.69%, 1.74%, 1.92%, 2.61%, and 4.35% respectively. These percentage values are fairly good within the margin of experimental, numerical and curve fitting errors.

The main purpose of these additional runs is to check that the numerical simulator produces over time, the normal profiles observed in the laboratory. The results in Figure 7.51 confirm this and hence the simulator can be used routinely for modeling laboratory immiscible displacement with some degree of confidence.

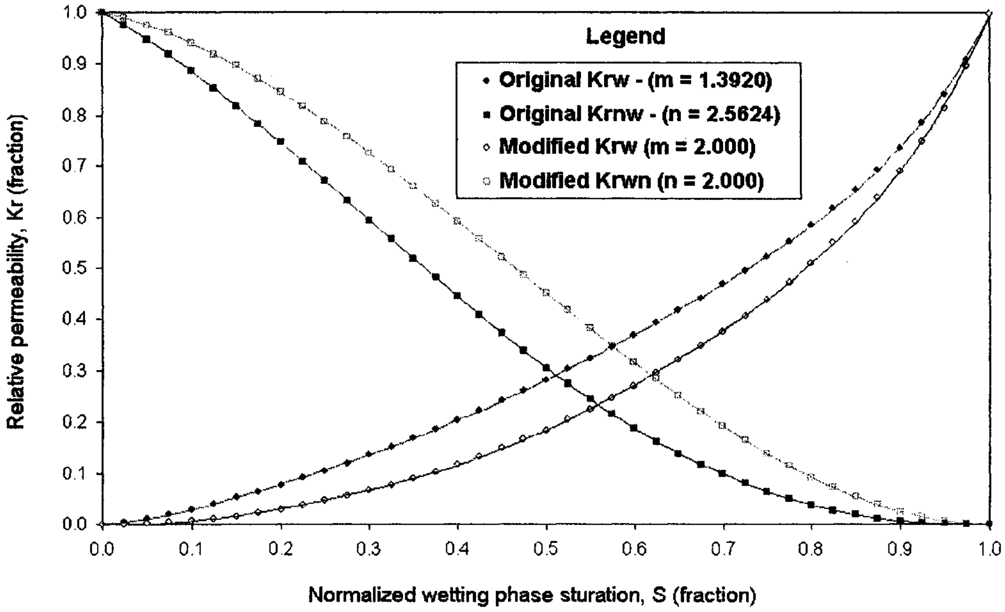


Figure 7.51 : Original and modified relative permebality curves for Data Group A

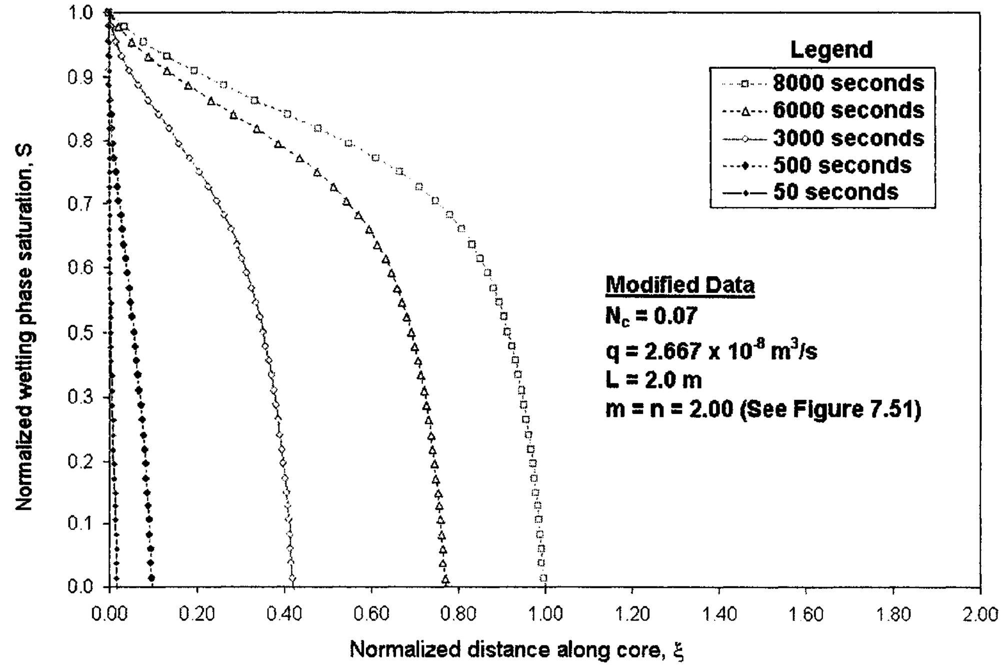


Figure 7.52 : Evolution of saturation profiles over time (Modified Data Group A)

7.3.3 Concluding Remarks on Numerical Verifications

From the results presented in Section 7.3.2, it can be concluded that the modified set of transport equations gives a good description of horizontal, two-phase porous media, cocurrent flow. Hydrodynamic effects can be neglected in cocurrent flow. Viscous and capillary coupling effects are also insignificant in cocurrent flow.

Sensitivity analysis shows that if the viscous factor in the parameter that controls the amount of viscous coupling is taken as the theoretically established value of 2 or below, then the viscous coupling effect is insignificant. Otherwise, the viscous coupling effect would have a greater effect on the physics of flow and might lead to unreasonable results. Due to the non-availability of a complete set of vertical flow data, no vertical flow simulation runs were carried out and no comparison was made with vertical flow experimental data.

Because the fractional flow concept breaks down in countercurrent flow situations, the numerical simulator cannot handle such a situation and hence no sensitivity analysis or numerical simulation studies were carried out for countercurrent flow situations. Prediction of saturation profiles over time confirms that the numerical simulator can be used routinely for cocurrent flow numerical studies.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1. CONCLUSIONS

By bearing in mind the objectives set out in Chapter 3, the following conclusions can be drawn based on the analytical and numerical simulation investigation results presented in this dissertation.

8.1.1. Conclusion Based on Analytical and Experimental Results

1) Expressions were developed for the quantification of interfacial coupling in two-phase porous media flow. These expressions were incorporated into the conventional Darcy equation. This leads to a modification of the Darcy equation for two-phase flow.
2) Acquired experimental data as well as experimental data available in the literature were used for experimental verification of the transport equations in (1).
3) The expression, $X\phi^2$, was developed for the quantification of the parameter that controls the amount of viscous coupling. The constant X was theoretically found to have a maximum value of 2 and the constant was experimentally found to be about 0.001 or less in order to effectively account for the effect of viscous coupling. On the other hand, the capillary coupling expression, $1-\phi$, developed by Bentsen was used in the experimental analysis.
4) Comparison of analytical and experimental results show that viscous coupling is very small based on the expression mentioned in (3) above, capillary coupling is larger and hydrodynamic effects can be neglected in horizontal, countercurrent flow. Both capillary and viscous coupling effects are shown to be very small in horizontal, cocurrent flow and can be neglected. No final conclusion can be made with regards to vertical flow. More experimental data will be required to a make final conclusion with regards to the applicability of the transport equation to vertical flow.
5) If can be concluded that interfacial coupling effects (capillary and viscous) exist in two-phase porous media flow. When transport equations are written in terms of effective permeability, the interfacial coupling effects should be included in the values determined using the expressions presented in this research. It can also be inferred that either the cocurrent or the countercurrent relative permeability values can be determined experimentally and used in reservoir simulation.
6) If the conventional experimentally determined countercurrent effective permeability values are used in simulating a countercurrent flow recovery process, then the recovery prediction would be correct. Also, if the conventional experimentally determined countercurrent effective permeability values are used in simulating a cocurrent flow recovery process, then the countercurrent effective permeability values would have to be divided by the interfacial coupling parameter, $\alpha_i = \alpha_{ci}.\alpha_{vi}$, in order to obtain a much more accurate recovery prediction. On the other hand, the effective permeability values would have to be multiplied by the interfacial coupling parameter if the conventional experimentally determined cocurrent effective permeability values are used in the numerical simulation of a countercurrent flow.

7) The formulation of the transport equations in (1) above does not account for a combined cocurrent and countercurrent recovery process. It only accounts for situations where each configuration of flow occurs separately.
8) The original objective of conducting cocurrent steady state and cocurrent unsteady state experiments was achieved. But the objective of conducting steady state countercurrent horizontal flow could not be achieved due to problems mentioned in Chapter 6. As a result of this, experimental data available in the literature were used in the analysis of countercurrent and cocurrent flow.

8.1.2. Conclusion Based on Numerical Simulation Results

9) The modified transport equations in (1) above were normalized and transformed to compact forms. The fractional flow equation was transformed into a Bentsen-type equation.
10) Numerical techniques and schemes were developed for the solution of the equations. The solutions were based on a fully implicit technique.
11) The numerical solution algorithm was used to develop a standalone numerical simulator with Java - for numerical verification of (1) above. Computer software, which is a well-structured combination of computer programs with a graphical user interface, was developed with the Java programming language for investigating interfacial coupling effects in porous media flow. The software is a numerical implementation of the solution of the modified set of transport equations given in Chapter 4. This software is referred to as the "interfacial-coupling simulator". The numerical implementation was based on the numerical schemes and algorithm developed in Chapter 5. The software enables the use of available data to verify the modified set of transport equations and carry out sensitivity analysis on the transport equations.
12) This research activity is one of the few efforts in which the Java programming language was employed for writing a numerical simulator. Most available commercial and research/laboratory numerical reservoir simulators were written with FORTRAN and/or C/C++ programming languages. So, the developed software can be regarded as one of the pioneer efforts in the use of Java for writing a numerical reservoir simulator.
13) From the results obtained from the numerical studies undertaken with the simulator, it can be concluded that the modified set of transport equations gives a good description of horizontal, two-phase, porous media, cocurrent flow.
14) Sensitivity analysis carried out with the numerical simulator shows that the viscous and capillary coupling effects are insignificant in cocurrent flow. Also, the hydrodynamic effect can be neglected in horizontal, two-phase, porous media, cocurrent flow.
15) Because the fractional flow concept breaks down in countercurrent flow situations, the numerical simulator cannot handle such a situation and hence no sensitivity analysis or numerical simulation studies were carried out for countercurrent flow situations.
16) Numerical simulation results confirm that the simulation can predict correctly expected patterns of saturation profiles over time. Hence, the simulator can be used routinely for cocurrent numerical simulation studies.
17) Numerical results of horizontal flow were used in the analyses. Due to the non-availability of a

complete set of vertical flow experimental data, no analysis was made for vertical flow.

8.2. RECOMMENDATIONS

Based on the outcome of this research, the following recommendations are made for further studies. These can be classified into recommendations that pertain to the experimental measurement system and those that pertain to numerical simulation studies.

8.2.1 Experimental

1) X-ray computed tomography-based equipment should be used as opposed to electrical-based measurements system for saturation measurements. This type of equipment gives a more accurate result. Also, such equipment makes use of a consolidated core, which is more representative of the type of porous media encountered in the field.
2) If the cost of the above is a deterrent, a better relationship for establishing saturation values from electrical measurements should be employed. For example, relationships for interfacial polarization frequency (IPF), lower critical frequency and dissipation factor (Su et al. (2000)); relationships between di-electric permittivity, saturation levels and other properties of the medium under investigation (Nguyen et al. (1999), West et. al. (2003) and Cosenza et al. (2003)); measurement relationships at very high frequencies (Bona et. al. (2001 and 2002); etc. should be investigated.
3) Also, if the present measurement system is going to be used in the future, other types of sensor/transducer and membranes/disc that enable accurate determination of wetting phase pressure with minimal or no error should be employed

8.2.2 Numerical

4) It is recommended that the software be used in immiscible displacement numerical studies. The software can be used to conduct cocurrent simulation studies.
5) The software source codes are open and available. So it is recommended that they can be customized and recompiled with a Java compiler (javac). Also, if the physics of the underlying flow configuration changes, the source codes can be adapted for this.
6) Due to the operating system-independence (architectural neutrality) of the Java programming language, the software can also be installed on any personal computer on which a Java virtual machine (JVM) is available without any recompilation of the source codes. JVMs for widely used computer operating systems (Unix, Windows, Linux, Solaris, Macintosh, etc.) are freely available from http://java.com/en/index.jsp. A minimum computer memory size of 128 Megabytes and a processor speed of 800 MHz are recommended.
7) All the classes that were developed were packaged into the **ics** package and compiled with Java 2 Standard Edition (J2SE - build 1.4.2_02). This version of **ics** was implemented in a full objected-oriented programming (OOP) paradigm. This version has some inherent OOP advantages but computation time might be too high when a higher saturation grid number (say 200 or more) is used for solving the equations. This is particularly due to Java array indexing overhead and the overhead of creating several objects during execution of the program. To overcome this problem, it is recommended that the rules of objected-oriented programming be relaxed to develop a partial "FORTRAN Style" analogue of the **ics** package. " FORTRAN Style

(FS) means all methods (procedures) are static, arrays are passed directly as arguments, and the data is accessed directly" (Markidis et al. (2002)). Hence, partial FS means, some methods are static, some arrays are passed directly as arguments, and some data is accessed directly while others are not. An implementation of the partial analogue will produce an improvement in the execution time as reported by Markidis et al. for a full FS analogue.

REFERENCES

- Adisoemarta, P.S. (2000). Detecting changes in shale water content through changes in electrical dissipation factor. Petroleum Society paper number 2000-32 presented at the Petroleum Society's international petroleum conference, Calgary, Alberta, Canada, 4-8 June 2000.
- Akin, S. (2001). Estimation of fracture relative permeabilities from unsteady state corefloods. J. Petrol. Sci. Eng. **30**, 1-14.
- Akin, S. (1999). Imbibition studies of low-permeable porous media. SPE paper number 54590 presented at the 1999 SPE western regional meeting, Anchorage, Alaska, 26-27 May 1999.
- Akin, S. and Demiral, B. (1998). Genetic algorithm for estimating relative permeabilities from displacement experiments. Computers and Geosciences. **24**(3), 251-258.
- Ali, L. and Barrufet, M.A. (2001). Using centrifuge data to investigate the effects of polymer treatment on relative permeability. J. Petrol. Sci. Eng. **29**, 1-16.
- Allen, F.R., Maddison, G.P. and Puckett, D.A. (1982). Theoretical and experimental studies of rate dependent two-phase immiscible flow. SPE paper number 10972 presented at the 1982 SPE annual technical conference and exhibition, New Orleans, Louisiana, April 2-4 1982.
- Al-Wadahi, M. Grader, A.S. and Ertekin, T. (2000). An investigation of three-phase counter-current flow using x-ray computerized tomography and neuro-simulation modeling. SPE paper number 63146 presented at presented at the 2000 SPE annual technical conference and exhibition, Dallas, Texas, October 1-4 2000.
- Ames, W.F. (1992). Numerical methods for partial differential equations. 3rd Edition. Academic Press, New York, US. 354-355.
- Amyx, J.W., Bass, D.M., Jr. and Whiting, R.L. (1960). Petroleum reservoir engineering. McGraw Hill Book Co., New York.
- Archer, J.S. and Wong, S.W. (1973). Use of a reservoir simulator to interpret laboratory waterflood data. SPE Reservoir Engineering. December 1973, **13**(6), 343-347.
- Aridor, Y., Factor, M. and Teperman, A. (2001). A distributed implementation of a virtual machine for Java. Concurrency: Practice and Experience. **13**(3), 221-244.
- Arulanandan, K. and Smith, S.S., (1973). Electrical dispersion in relation to soil structure. Journal of the Soil Mechanics and Foundations Division. **December** 1973, 1113-1133.
- Ayub, M. (2000). Experimental testing of interfacial coupling phenomena in two-phase flow. A thesis submitted to the Department of Civil of and Environmental Engineering of the University of Alberta in partial fulfillment for the award of the degree of Doctor of Philosophy in Petroleum Engineering.
- Ayub, M. and Bentsen, R.G. (2004). Experimental testing of interfacial coupling in two-phase flow in porous media. Petroleum Sci. & Tech. In press.
- Ayub, M. and Bentsen, R.G. (2001). An apparatus for simultaneous measurement of relative permeability and dynamic capillary pressure. Petroleum Sci. & Tech. **23**(9 &10), 1129-1154.
- Ayub, M. and Bentsen, R.G. (2000). Measurement of dynamic saturation profiles. JCPT. **September** 2000, **39**(9), 54-61.

- Ayub, M. and Bentsen, R.G. (1999). Interfacial viscous coupling: a myth or reality? J. Petrol. Sci. Eng. **23**, 13-26.
- Babchin, A., Yuan, J. and Naser, T. (1998). Generalized phase mobilities in gravity drainage processes. Petroleum Society paper number 98-09 presented at the 49th annual technical meeting of the Petroleum Society, Calgary, Alberta, Canada, June 8-10, 1998.
- Babchin, A. and Yuan, J. (1997). On the capillary coupling between two phases in a droplet train model. Transport in Porous Media. **26**, 225-228.
- Bachmat, Y. and Bear, J. (1986). Macroscopic modeling of transport phenomena in porous media. 1: The continuum approach. Transport in Porous Media. **1**, 213-240.
- Bacri, J., Chaouche, M. and Salin, D. (1990). Modele simple de permeabilities croisees. C. R. Acad. Sci., Paris. **311**, 591-596, Series II.
- Bail, P.T. and Marsden, S.S., (1957). Saturation distribution in a linear system during oil displacement. Producers Monthly. **21**(8), 22-32.
- Baker, L.E. (1988). Three-phase relative permeability correlations. SPE paper number 17369 presented at the 1988 SPE/DOE enhanced oil recovery symposium, Tulsa, Oklahoma, April 1988.
- Bates, R.H.T., Garden, K.L. and Peters, T.M. (1983). Overview of computerized tomography with emphasis on future developments. Proc. IEEE. **71**(3). 356-372.
- Batycky, J.P., McCaffery, F.G., Hodgins, P.K. and Fisher, D.B. (1981). Interpreting relative permeability and wettability from unsteady-state displacement measurements. SPE Journal. **21**(3), 296-308.
- Bear, J. (1972). Dynamics of fluids in porous media. American Elsevier Publishing Company, Inc., New York.
- Bear, J. and Bachmat, Y. (1986). Macroscopic modeling of transport phenomena in porous media. 2: Application to mass, momentum and energy transport. Transport in Porous Media. **1**, 241-269.
- Bedford, A. and Drumheller, D.S., (1983). Recent advances - theories of immiscible and structured mixtures. Int. J. Engg. Sci. **21**(8), 863-960.
- Bèkri, S., Vizika, O., Thovert, J.F. and Adler, P.M. (2001). Binary two-phase flow with phase change in porous media. Int. J. Multiphase Flow. **27**, 477-526.
- Bell, D. and Parr, M. (1999). Java for students. (1999) Second Edition. Prentice Hall, London, UK. xvii-xix.
- Bentsen, R.G. (2003a). The role of capillarity in two-phase flow through porous media. Transport in Porous media. **51**, 103-112.
- Bentsen, R.G. (2003b). Comment on capillary coupling. Transport in Porous media. **51**, 371.
- Bentsen, R.G. (2003c). Interfacial coupling in vertical, two-phase flow through porous media. Paper Submitted to J. Petrol. Sci. Eng.
- Bentsen, R.G. (2001). The physical origin of interfacial coupling in two-phase flow through porous media. Transport in Porous Media. **44**, 109-122.
- Bentsen, R.G. (1998a). Influence of hydrodynamic forces and interfacial momentum transfer on the flow of two immiscible phases. J. Petrol. Sci. Eng. **19**, 177-190.

- Bentsen, R.G. (1998b). Effect of momentum transfer between fluid phases on effective mobility. J. Petrol. Sci. Eng. **21**, 27-42.
- Bentsen, R.G. (1997). Impact of model error on the measurement of flow properties needed to describe flow through porous media. Revue de l'institut francais du petrole. May-June, 1997, **52**(3), 299-315
- Bentsen, R.G. (1994a). Effect of hydrodynamic forces on capillary pressure and relative permeability. Transport in Porous Media. **17**, 121-132.
- Bentsen, R.G. (1994b). Effect of hydrodynamic forces on the pressure-difference equation. Transport in Porous Media. **17**, 133-144.
- Bentsen, R.G. (1992). Construction and experimental testing of a new pressure-difference equation. AOSTRA J. Res. **8**, 159-168.
- Bentsen, R.G. (1985). A new approach to instability theory in porous media. SPE Journal. **October** 1985, 765-779.
- Bentsen, R.G. (1978). Conditions under which the capillary term can be neglected. JCPT. **October - December** 1978, **17**(4), 25-30.
- Bentsen, R.G. (1976). Scaled fluid-flow models with permeabilities differing from that of the prototype. JCPT. **July - September** 1976, **15**(3), 46-52.
- Bentsen, R.G. and Manai, A. A. (1993). On the use of conventional cocurrent and countercurrent effective permeabilities to estimate the four generalized permeability coefficients which arise in coupled, two-phase flow. Transport in Porous Media. **11**, 243-262.
- Bentsen, R.G. and Manai, A. A. (1991). Measurement of cocurrent and countercurrent relative permeability curves using the steady-state method. AOSTRA J. Res. **7**, 169-181.
- Bentsen, R.G. and Saeedi, J. (1981). Liquid-liquid immiscible displacement in unconsolidated porous media. JCPT. **January - March** 1981, **20**(1),93-103.
- Berg, C.R. (1995). A simple, effective-medium model for water saturation in porous rocks. Geophysics. **60**(4), 1070-1080.
- Bhatnagar, P.L., Gross E.P. and Kroot, M. (1954). A model for collision processes in gases. 1. Small amplitude processes in charged and neutral one-component systems. Phys. Rev. **94**(5), 511-525.
- Blair, P.M. (1960). Calculation of oil displacement by countercurrent water imbibition. SPE paper number 813 presented at the 1960 SPE 4th Biennial secondary symposium, Wichita Falls, Texas, May 2-3 1960.
- Blunt, M.J. (2000) An empirical model for three-phase relative permeability. SPE Journal. **December** 2000. SPE paper number 56474 (final number 67950) originally presented at the 1999 SPE annual technical conference and exhibition, Houston, Texas, Oct. 1999.
- Boisvert, R.F., Moreira, J., Philippsen, M. and Pozo, R. (2001). Java and numerical computing. Computing in Science and Engineering (CiSE), **March/April** 2001, 18-24.
- Boisvert, R.F., Dongarra, J.J., Pozo, R., Remington, K.A. and Stewart, G.W. (1998). Developing numerical libraries in Java. Concurrency: Practice and Experience. **10**(11-13), 1117-1129.
- Bolt, G.H. and Groenevelt, (1969). Coupling phenomenon as a possible cause for non-Darcian behavior of water in soil. Bull. I.A.S.H. Vol. **14**(2), 17-26.

- Bona, N., Ortenzi, A. and Capaccioli, S. (2002). Advances in understanding the relationship between rock wettability and high-frequency dielectric response. J. Petrol. Sci. Eng. **33**, 87-99.
- Bona, N., Rossi, E. and Capaccioli, S. (2001). Electrical measurements in the 100 Hz to 10 GHz frequency range for efficient rock wettability determination. SPE Reservoir Engineering. **March** 2001,80-88.
- Bonachea, D., Dickens, P. and Thakur, R. (2001). High-performance file I/O in Java: Existing approaches and bulk I/O extensions. Concurrency and Computation: Practice and Experience. **13**(8-9), 713-736.
- Bourbiaux, B.J. and Kalaydjian, F.J., (1990). Experimental study of cocurrent and countercurrent flows in natural porous media. SPE Reservoir Engineering Journal. **August** 1990, 361-368.
- Bowen, R.M., (1982). Compressible porous media models by use of the theory of mixtures. Int. J. Eng. Sci. **20**(6), 697-735.
- Bowen, R.M., (1980). Incompressible porous media models by use of the theory of mixtures. Int. J. Eng. Sci. **18**, 1129-1148.
- Bryant, S. and Blunt, M.J. (1992). Prediction of relative permeability in simple porous media. Phys. Rev. A. **46**, 2004.
- Buckley, S.E and Leverett, M.C. (1942). Mechanism of fluid displacement in sands. Trans. AIME. **146**, 107-116.
- Bull, J.M., Smith, L.A., Westhead, M.D., Henty, D.S. and Davey, R.A. (2000). A benchmark suite for high performance Java. Concurrency and Computation: Practice and Experience. **12**(6), 275-388.
- Burdine, N.T. (1953). Relative permeability calculations from pore size distribution data. Trans. IME. **198**, 71.
- Butler, R.M. (1994). Horizontal well for the recovery of oil, gas and bitumen. Petroleum Society Monograph number **2**. Chapter 11: Thermal Recovery Using Horizontal Well. 169-199.
- Cardwell, Jr. W.T. (1959). The meaning of the triple value in non-capillary Buckley-Leverett theory. Trans. AIME. **216**, 271.
- Carr, A.H. and Christie, M.A. (1983). Controlling numerical diffusion in reservoir simulation using flux corrected transport. SPE paper number 12235 presented at the 1983 SPE reservoir simulation symposium, San Francisco, California, Nov. 15-18, 1983.
- Chatenever, A. and Calhoun, J.C., Jr. (1952). Visual examinations of fluid behavior in porous media - Part 1. Trans. AIME.**195**, 149-156.
- Chavent, G., Cohen, G. and Espy, M. (1980). Determination of relative permeability and capillary pressure by an automatic adjustment method. SPE paper number 9237 presented at the SPE 55th annual fall meeting, Dallas, Texas, September 1980.
- Chen, J., Miller, M.A. and Sepehrnoori, K. (1995). Theoretical investigation of countercurrent imbibition in fractured reservoir matrix blocks. SPE paper number 29141 presented at the SPE 13th symposium on reservoir simulation, San Antonio, Texas, February 1995.

- Chen, H., Chen, S. and Mattaeus, W. H. (1992). Recovery of the Navier-Stokes equation using a Lattice-Gas Boltzmann method. Phys. Rev. A. Europhys. **45**, 5539-5542.
- Childs, E.C. (1969). An introduction to the physical basis of soil water phenomena. John Wiley and Sons Ltd., London, UK.
- Cheng, P. and Wang, C.Y., (1996). A multiphase mixture model for multiphase, multicomponent transport in capillary porous media: II. Numerical simulation of the transport of organic compounds in the subsurface. Int. J. Heat Mass Transfer. **39**(17), 3619-3632.
- Chierici, G. L. (1994). Principles of petroleum reservoir engineering, volume 2. Translated from the Italian version by Westaway, P.J. Springer-Verlag, Berlin, 3.
- Cil, M., Reis, J.C., Miller, M.A. and Mistra, D. (1998). An examination of countercurrent capillary imbibition recovery from single matrix blocks and recovery predictions by analytical matrix/fracture transfer functions. SPE paper number 49005 presented at the 1998 SPE annual technical conference and exhibition, New Orleans, Louisiana, September 27-30 1998.
- Civan, F. and Donaldson, E.C. (1987) Relative permeability from unsteady-state displacements: An analytical interpretation. SPE paper number 16200 presented at the 1987 SPE production operations symposium, Oklahoma City, Oklahoma, March 8-10 1987.
- Collins, R.E. (1961). Flow of fluids through porous materials. Reinhold Publishing Corporation, New York, 139-169.
- Cosenza, P., Camerlynck, C. and Tabbagh, A. (2003). Differential effective medium schemes for investigating the relationship between high-frequency relative dielectric permittivity and water contents of soils. Water Resources Research. **September** 2003. **39(9)**, 1:1-13.
- Craig, F.F., Jr. (1971). The reservoir engineering aspects of waterflooding. Monograph Vol. **3** of the SPE of AIME.
- Dana, E. and Skoczylas, F. (2002). Experimental study of two-phase flow in three sandstones. II. Measuring relative permeabilities during two-phase steady-state experiments. Int. J. Multiphase Flow. **28**, 1965-1981
- Davis, Jr., L.A., (1980). VHF electrical measurement of saturation in laboratory floods. SPE Paper number 8847 presented at the 1980 1st joint SPE/DOE symposium on enhanced oil recovery, Tulsa, Oklahoma, April 20-23, 1980.
- de Gennes, P.G. (1983). Theory of slow biphasic flows in porous media. Ph. Ch. Hydr. **4**, 175-185.
- de Groot, S.R., (1963).Thermodynamics of Irreversible Processes. North-Holland Publishing Co. Amsterdam.
- de Groot, S.R. and Mazur, P. (1963). Non-equilibrium thermodynamics. North-Holland Publishing, Amsterdam.
- de la Cruz and Spanos, T.J.T. (1983). Mobilization of oil ganglia. AIChE J. **29**(5), 854-858.
- de Wiest, (1969). Flow through porous media. Academic Press, New York, USA.
- del Rio P.J.A. and de Harro, M.L. (1992). Extended irreversible thermodynamics as a framework for transport phenomena in porous media. Transport in Porous Media. **9**, 207-221.
- Douglas, J., Jr., Blair, P.M. and Wagner, R.J. (1958). Calculation of linear waterflood behaviour including the effects of capillary pressure. Trans. AIME. **213**, 96-102.

- Douglas, J., Jr., Darlow, B.L., Wheeler, M. and Kendall, R. (1979). Self-adaptive Galerkin methods for one-dimensional, two-phase immiscible flow. SPE paper number 7679 presented at the 1979 SPE of AIME fifth symposium on reservoir simulation, Denver, Colorado, Feb. 1979.
- Douglas, J., Jr., Peaceman, D.W. and Rachford, H.H., Jr. (1959). A method for calculating multi-dimensional displacement. Trans. AIME. **216**, 297-308.
- Douglas, J., Jr., Pereira, F. and Yeh, L.-M. (2000). A locally conservative Eulerian-Lagrangian numerical method and its application to nonlinear transport in porous media. Computational Geosciences. 4, 1-40.
- Drumheller, D.S., (1978). The theoretical treatment of a porous solid using a mixture theory. Int. J. Solids Structures. **14**, 441-456.
- Dullien, F.A. L. (1992). Porous media: Fluid transport and pore structure. Second Edition. Academic Press Limited, San Diego. 333-337,341-380.
- Ellingson, W.A., Roberts, R.A., Vannier, M.W., Ackerman, J.L., Sawicka, B.D., Gronermeyer, S. and Kriz, R.J. (1987). Recent developments in nondestructive evaluation of structural ceramics. Argonne National Laboratory. ANL/FE-87-2.
- Fassihi, M.R. (1989). Estimation of relative permeability from low rate, unsteady-state tests - A simulation approach. JCPT. **May - June**, 1989, **28**(3), 29-38.
- Fatt, I. and Dyksra, H. (1951). Relative permeabilities studies. Trans. AIME. **192**, 249-256.
- Fayers, F.J. and Sheldon, J.W. (1959). The effect of capillary pressure and gravity on two-phase fluid flow in a porous medium. Trans. AIME. **216**, 147-155.
- Fenwick, D.H. and Blunt, M.J. (1998a). Network modeling of three-phase flow in porous media. SPE Journal. **March** 1998, 86-97.
- Fenwick, D.H. and Blunt, M.J. (1998b). Three-dimensional modeling of three-phase imbibition and drainage. Advances in Water Resources. 25, 121-143.
- Ferreol, B. and Rothman, D.H., (1995). Lattice-Boltzmann simulation of flow through Fontainebleau sandstone. Transport in Porous Media. **20**, 3-20.
- Firoozabadi, A. and Aziz, K. (1991). Relative permeabilities from centrifuge data. JCPT. September - October 1991, **30**(5), 33-42.
- Flanagan, D. (1997). Java in a nutshell: A desktop quick reference. Second Edition. O'Reilly and Associates, Inc., Sebastopol, California, US. 3-9.
- Friedman-Hill E.J. (2001). Java: Your visual blueprint for building portable Java programs. Hungry Minds, Inc., New York, US. 2-3.
- Frisch, U., Hasslacher, B., d'Humières, D., Lallemand, P., Pomeau, Y. and Rivet, J.P. (1987). Lattice-gas hydrodynamics in two and three dimensions. Complex Syst. **1**, 649-708.
- Frisch, U., Hasslacher, B. and Pomeau, Y., (1986). Lattice-gas automata for the Navier-Stokes equation. Phys. Rev. Lett. **56**(14), 1505-1508.
- Gao, Y. and Sharma, M.M., (1994a). A LGA model for dispersion in heterogeneous porous media. Transport in Porous Media. **17**, 19-32.
- Gao, Y. and Sharma, M.M., (1994b). A LGA model for fluid flow in heterogeneous porous media. Transport in Porous Media. **17**, 1-17.

- Geffen, T.M. and Gladfelter, R.E. (1952). A note on the X-ray absorption method of determining fluid saturations in cores. Trans. AIME. **195**, 322-323.
- Geffen, T.M., Owens, W.W., Parrish, D.R. and Morse, R.A. (1951). Experimental investigation of factors affecting laboratory relative permeability measurements. Trans. AIME. **192**, 99-110.
- Getov, V., von Laszewski G., Philippsen, M. and Foster, I. (2001). Multiparadigm communications in Java for grid computing. Communications of the ACM. October 2001. **44** (10), 118-125.
- Gilboy, W.B. (1984). X- and γ-ray tomography in NDE applications. Nuclear Instrum. Methods Phys. Res. **221**, 193-200.
- Goode, P.A. and Ramakrishnan, T.S. (1993). Momentum transfer across fluid-fluid interfaces in porous media: a network model. AIChE J. **39**(7), 1124-1134.
- Goode, P.A. (1991). Momentum transfer across fluid-fluid interfaces in porous media. Ph.D. thesis, Heriot-Watt University, U.K.
- Gosling, J., Joy B. Steele, G. and Bracha, G. (2000). The Java language specification. Second edition. Addison-Wesley, Boston, MA, USA.
- Graham, J.W. and Richardson, J.G. (1959). Theory and application of imbibition phenomena in recovery of oil. Trans. AIME. **216**, 377-381.
- Gray, W.G., (1975). A derivation of the equations for multi-phase transport. Chem. Eng. Sci. **30**, 229-233.
- Gray, W.G. and Lee, P.C.Y. (1977). On the theorems for local volume averaging of multiphase systems. Int. J. Multiphase Flow. **3**, 333-340.
- Greenkorn, R. A. (1983). Flow phenomena in porous media. Marcel Dekker, New York, USA.
- Guevara-Jordan, J.M. and Glimm, J. (1997). A mixed finite element method for the Hele-Shaw cell equations. Computational Geosciences. **1**(1), 35-58.
- Gutman, S., (1990). Lattice gas methods in porous media computations. Annual AIChE meeting (Rock/Fluid Interaction in Petroleum Reservoirs I, Chicago, Illinois, Nov. 11-16, 1990.
- Guo, Y. and Vatne, K.O. (1993). Use of a new generation re-circulation system for steady state relative permeability measurement. Proceedings of the of the 7th European IOR symposium, Moscow, 1993.
- Hardy, J., de Pazzis, O. and Pomeau, Y. (1976). Molecular dynamics of classical lattice gas: transport properties and time correlation functions. Phys. Rev. Ser. A, **13**, 19-47-1961.
- Hagoort, J. (1980). Oil recovery by gravity drainage. SPE Journal. **June** 1980, 139-150.
- Handy, L.L. and Hardley, G. (1956). A theoretical and experimental study of the steady state capillary end effect. Paper 707-G, presented at the 31st annual fall meeting of the SPE, Los Angeles (Oct. 14-17, 1956).
- Hassanizadeh, M. and Gray, W.G. (1980). General conservation equations for multi-phase systems: 3. Constitutive theory for porous media flow. Advances in Water Resources. **3**, 25-40.

- Hassanizadeh, M. and Gray, W. G. (1979a). General conservation equations for multi-phase systems: 2. Mass, momenta, energy and entropy equations. Advances in Water Resources **2**, 191-203.
- Hassanizadeh, M. and Gray, W. G. (1979b). General conservation equations for multi-phase systems: 1. Averaging procedure. Advances in Water Resources **3**, 25-40.
- Hassler, G.L. and Brunner, E. (1945). Measurement of capillary pressures in small core samples. Trans. AIME. **160**, 114-123.
- Hassler, G.L., Rice, R.R. and Leeman, E.H. (1936). Investigations on the recovery of oil from sandstones by gas drive. Trans. AIME. **118**, 116-137.
- Heaviside, J. (1991). Interfacial phenomena in petroleum recovery. Surfactant science series volume 36. Chapter 10: Measurement of Relative Permeability. Edited by Morrow, N. R. Marcel Dekker Inc. New York, 377-411.
- Heaviside, J., Brown, E. and Gamble. I.J.A. (1988). Relative permeability for intermediate wettability reservoir. SPE paper number 16968 presented at the 1988 SPE annual technical conference and exhibition, Houston, Texas, October 2-5 1988.
- Heiba, A.A., Davis, H.T. and Scriven, L.E. (1984). Statistical network theory of three-phase relative permeabilities. SPE/DOE paper number 12690 presented at the 1984 SPE/DOE improved oil recovery symposium, Tulsa, Oklahoma, April 15-18 1984.
- Henderson, G.D., Danesh, A., Al-kharusi, B. and Tehrani, D. (2000). Generating reliable gas condensate relative permeability data used to develop a correlation with capillary number. J. Petrol. Sci. Eng. **25**, 79-91.
- Hicklin, J., Moler, C. Webb, P. Boisvert, R.F., Miller, B., Pozo, R. Remington, K. (1998). The Java matrix package - Jama. http://math.nist.gov/javanumerics/jama/
- Hicks, P.J. (1996). X-ray computer-assisted tomography for laboratory core studies. Technology today series. JPT. **48**(12), 1120-1122.
- Hiller, D. and Ermert, H. (1984). System analysis of ultrasonic reflection mode computerized tomography. IEEE Trans. Sonic Ultrasonic, SU. **31**(4), 240-250.
- Hirasaki, G.J., Rohan, J.A., and Dudley, J.W. II. (1992). Interpretation of oil/water relative permeabilities from centrifuge displacements. SPE paper number 24879 presented at the 1992 SPE 67th annual technical conference and exhibition, Washington, DC, Oct. 4-7, 1992.
- Hoffmann, J.D. (2001). Numerical methods for engineers and scientists. Second edition. Revised and expanded. Marcel Dekker Inc. New York, New York, US. 619-621.
- Honarpour, M., Koederitz, F.L. and Harvey, A.H. (1986). Relative permeability of petroleum reservoirs. Chapters 1, 2 and 3. CRC Press Inc. Boca Raton, 1-102.
- Honarpour, M. and Mahmood, S.M. (1988). Relative-permeability measurements: An overview. Technology today series. JPT. **August** 1988, 963-966.
- Hovanessian, S.A. and Fayers, F.J. (1961). Linear waterflood with gravity and capillary effects. Soc. Pet. Eng. Journal. March 1961. **1**, 32-36.
- Hove, A., Ringen, J.K. and Read, P.A. (1985). Visualization of laboratory corefloods with the aid of computerized tomography of X-rays. SPE paper number 13654 presented at the 1985 SPE California regional meeting, Bakersfield, California, March 27-29, 1985.

- Huan, G. and Shen, P. (1982). A method for calculating unsteady state oil-water relative permeability. Journal of petroleum Exploration and Development. **2**, 52.
- Islam, M.R. and Bentsen, R.G. (1986). A dynamic method for measuring relative permeability. JCPT. **January-February** 1986, 39-50.
- Islam, M.R. and Bentsen, R.G. (1987). Effect of different parameters on two-phase relative permeability. AOSTRA J. Res. **3**, 69-90.
- Jensen, O. K. (1989). A review of reservoir simulation techniques and considerations on future developments. In proceedings of the Conference on the Mathematics of Oil Recovery organized by the Institute of Mathematics of Oil Recovery and its Applications in association with Society of Petroleum Engineers (SPE), held at Robinson College, Cambridge in July 1989, 1-21.
- Johnson, E.F., Bossler, D.P. and Naumann, V.O. (1959). Calculation of relative permeability from displacement experiments. Trans. AIME. **216**, 370-372.
- Jones, S.C. and Roszelle, W.O. (1978). Graphical techniques for determining relative permeability from displacement experiments. JPT. **May** 1978, **30**(5), 807-817.
- Kadanoff, L.P., McNamara, G.R. and Zanetti, G. (1989). From automata to fluid flow: comparisons of simulation and theory. Physical Review A. **40**(8), 4527-4541.
- Kalaydjian, F. (1990). Origin and quantification of coupling between relative permeabilities for two-phase flows in porous media. Transport in Porous Media **5** (3), 215-229.
- Kalaydjian, F. (1987). A macroscopic description of multiphase flow in porous media involving space-time evolution of fluid/fluid interface. Transport in Porous Media. **2**, 537-552.
- Kalaydjian, F. and Legait, B. (1987). Permeabilities relatives couplees dans des ecoulements en capillaires et en milieux poreux. C.R. Acad. Sci. Paris, 304, series II, 1035-1038.
- Kalaydjian, F. and Marle, C-M. (1989). Microscopic flow and generalized Darcy's equation. In proceedings of the Conference on the Mathematics of Oil Recovery organized by the Institute of Mathematics of Oil Recovery and its Applications in association with Society of Petroleum Engineers (SPE), held at Robinson College, Cambridge in July 1989, 495-514.
- Kalaydjian, F. and Marle, C.-M. (1987). Thermodynamic aspects of multiphase flow in porous media. 2nd Inst. Français Du Pétrole Explor. Res. Conf. (Migration of Hydrocarbons in Sedimentary Basins), Carcans, France, June 15-19, 1987, 513-531.
- Kantzas, A., Nikakhtar, B., Ruth, D. and Pow, M. (1995). Two-phase relative permeabilities using the ultracentrifuge. JCPT. **September** 1995, **34**(7), 58-63.
- Katchalsky, A. and Curran, P.F. (1967). Non-equilibrium thermodynamics in biophysics. Harvard University Press, Cambridge, Massachusetts.
- Kerig, P.D. and Watson, A.T. (1987). A new algorithm for estimating relative permeability from displacement experiments. SPE Reservoir Engineering Journal. **February** 1987, 103-112.
- Kerig, P.D. and Watson, A.T. (1986). Relative permeability estimation from displacement experiments: An error analysis. SPE Reservoir Engineering Journal. **March** 1986, 175-182.
- Khuzhayorov, B.Kh. and Burnashev, V.F. (2001). Modeling the multiphase flow of oil-gas-condensate system in porous media. J. Petrol. Sci. Eng. **29**, 67-82.

- Kielmann, T., Hatcher, P. Bougs L., and Bal, H.E. (2001). Enabling Java for high-performance computing. Communications of the ACM. October 2001. **44** (10), 110-117.
- Kimbler, O.K. and Caudle, B.H. (1957). New technique for study of fluid flow and phase distribution in porous media. Oil & Gas J. **55**(50), 85-88.
- King, P.R., (1996). Upscaling permeability: Error analysis for renormalization. Transport in Porous Media. **23**, 337-354.
- King, M.J. Narayanan, K.R. and Falzone, J. (1990). Advances in centrifuge methodology for core analysis. SCA paper number 9011 presented at the 1990 annual technical conference of the SCA.
- Klute, A. (1967). Notes on flow water in unsaturated soils. Paper presented at the M.I.T. summer session on ground water hydrology and flow through porous media.
- Kraszewski, A. (1996). Microwave aquametry - Electromagnetic wave interaction with water-containing materials. IEEE Press New York.
- Kulkarni, K.N and Datta-Gupta, A. (2000). Estimating relative permeability from production data: a streamline approach. SPE Journal. **December** 2000, 402-411.
- Küntz, M., van Mier, J. and Lavallée (2001). A lattice gas automaton simulation of the nonlinear diffusion equation in unsaturated porous media. Transport in Porous Media. **43**, 289-307.
- Laird, A.D.K. and Putman, J.A. (1959). Three component saturation in porous media by X-ray techniques. Trans. AIME. **216**, 216-220.
- Laird, A.D.K. and Putman, J.A. (1951). Fluid saturation in porous media by X-ray technique. Trans. AIME. **192**, 275-284.
- Langaas, K. and Papatzacos (2001). Numerical investigations of the steady state relative permeability of a simplified porous media. Transport in Porous Media. **45**, 241-266.
- Lelièvre, R.F. (1966). Etude d' écoulements disphasiques permanents à contre-courants en milieu poreux - Comparaison avec les écoulements de même sens (in French). Ph.D. Thesis, University of Toulouse, France.
- Lerdahl, T.R., Øren P.-E. and Bakke, S. (2000). A predictive network model for three-phase flow in porous media. SPE/DOE paper number 59311 presented at the 2000 SPE/DOE improved oil recovery symposium, Tulsa, Oklahoma, April 3-5 2000.
- Leverett, M.C. (1941). Capillary behaviour in porous solids. Trans. of AIME. **142**,152-169.
- Liang, Q. and Lohrenz, J. (1994). Dynamic method of measuring coupling coefficients of transport equations of two-phase flow in porous media. Transport in Porous Media. **15**, 71-79.
- Lindholm, T. and Yellin, F. (1999). The Java virtual machine specification. Second edition. Addison-Wesley, Boston, MA, USA.
- Longeron, D., (1987). Study of the influence of thermodynamic parameters on two-phase flow in porous media. 2nd Inst. Francais Du Petrole Explor. Res. Conf. (Migration of Hydrocarbons in Sedimentary Basins), Carcans, France, June 15-19, 1987, 257-279.
- MacCuaig, N., Tajuddin, A.A. and Gilboy, W.B. (1986). Industrial tomography using a positive sensitive carbon fibre anode proportional counter. Nuclear Instrum. Method Phys. Res. **A242**, 620-625.

- Maini, B. (1998). Is it futile to measure relative permeability for heavy oil reservoirs? JCPT. **April** 1998, **37**(4), 56-62.
- Maini, B. and Batycky, J.P. (1985). Effect of Temperature on heavy oil/water relative permeabilities in horizontally and vertically drilled core plugs. JPT. **August** 1985, **37** (9), 1500-1510.
- Maini, B., Coskuner, G. and Jha, K (1990). A comparison of steady state and unsteady states relative permeabilities of viscous oil and water in Ottawa sand. JCPT. **March-April** 1990, **29**(2), 72-77.
- Manai, A. A. (1991). The measurement of cocurrent and countercurrent relative permeabilities and their use to estimate the generalized relative permeabilities. A thesis submitted to the Department of Mining, Metallurgical and Petroleum Engineering of the University of Alberta in partial fulfillment for the award of the degree of Master of Science in Petroleum Engineering.
- Mani, V. and Mohanty, K.K. (1998). Pore-level network modeling of three-phase capillary pressure and relative permeability. SPE Journal. **September** 1998, 238-248.
- Manz, B. and Gladden, L.F. (1999). Flow and dispersion in porous media: Lattice-Boltzmann and NMR studies. AIChE Journal. **45**(9), 1845-1854.
- Malevanets, A. and Yeomans, J.M. (2000). Modeling viscous drag in binary fluid mixtures. Computer Physics Communications. **127(1)**, 105-112.
- Markidis, S., Lapenta, G. and VanderHeyden, W.B. (2002). Parsek: Object oriented particle-in-cell implementation and performance issues. In proceedings of the Joint ACM Java Grande - ISCOPE 2002 Conference, November 3-5, 2002, Seattle, Washington, USA. 141-147.
- Maximenko, A. and Kadet, V.V. (2000). Determination of permeabilities using the network models of porous media. J. Petrol. Sci. Eng. **28**, 145-152.
- McEwen, C.R. (1959). A numerical solution of the linear displacement equation with capillary pressure. Trans. AIME. **216**, 412-415.
- Moreira, J.E., Midkiff, S.P., Gupta M., Pedro V. Artigas, Peng Wu, and George Almasi (2001). The NINJA Project. Communications of the ACM. October 2001. **44** (10), 102-109.
- Morelli, R. (2000). Java, Java, Java! Object-oriented problem solving. Prentice-Hall Inc., Upper Saddle River, New Jersey, US. 4-14.
- Morel-Seytoux, H.J. (1969). Chapter 11: Introduction to flow of immiscible liquids in porous media. Flow through porous media: A compilation of several contributions on the topics of porous media studies. Academic Press New York, 456-516.
- Muccino, J. Gray. W. and Ferrand, L. (1998). Towards an improved understanding of multiphase flow in porous media, Rev. Geophy. **36**(3), 401-422.
- Munkvold, F.R. and Torsaeter, O. (1990). Relative permeability from centrifuge and unsteady sate experiments. SPE paper number 21103 presented at the 1990 SPE Latin American petroleum engineering conference, Rio de Janeiro, Brazil, Oct. 14-19, 1990.
- Murdoch, A.I. and Kowalski, S.J., (1992). On fluid-fluid coupling within porous media: A mixture theoretical approach based upon molecular considerations. Transport in Porous Media. **8**, 47-70.

- Muskat, M., Wyckoff, R.D., Botset, H.G. and Meres, M.W. (1937). Flow of gas-liquid mixtures through sands. Trans. AIME. **123**, 69-96.
- Muskat, M. and Meres, M.W. (1936). The flow of heterogeneous fluids through porous media. Physics. **7**, 346-363.
- Naar, J., Wygal, R.J. and Henderson, J.H. (1962). Imbibition relative permeability in unconsolidated porous media. SPE Journal. **March** 1962, 13-17.
- Narasimhan, T.N. (1980). A note on volume averaging. Advances in Water Resources, **3**, 35-139.
- Nasr, T.N., Law, D.H.S., Golbeck, H. and Korpany, G. (2000). Counter-current aspect of the SAGD process. JCPT. **January** 2000, **39**(1), 41-47.
- Nguyen, B. -L., Geels, A.M., Bruining, J. and Slob, E.C. (1999). Calibration measurements of dielectric properties of porous media. SPE Journal. **December** 1999, **4**(4), 353 -359.
- Norel, G. (1964). Absorption X des milieux poreux satures, application a la mesure des saturation et des porosites. Inst. Fr. Petrol. Ref. 10649.
- Nordtvedt, J.E., Watson, A.T., Mejia, G. and Yang, P. (1993). Estimation of capillary pressure and relative permeability functions from centrifuge experiments. SPE Reservoir Engineering. **November** 1993, 292-298
- O'Meara, D.J., Jr. and Crump, J.G. 1985. Measuring capillary pressure and relative permeability in a single centrifuge experiments. SPE paper number 14419 presented at the 1985 SPE 60th annual technical conference and exhibition, Las Vegas, Nevada, September 22-25, 1985.
- O'Meara, D.J., Jr. and Leas, W.O. 1983. Multiphase relative permeability measurement using an automated centrifuge. SPE paper number 12128 presented at the 1983 SPE 58th annual technical conference and exhibition, San Francisco, California, Oct. 5-8, 1983.
- Onsager, L. (1931a). Reciprocal relations in irreversible processes-I. Physical Review. **37**, 405-426.
- Onsager, L. (1931b). Reciprocal relations in irreversible processes-II. Physical Review. **38**, 2265-2279.
- Øren, P.E. Bakke, S. and Arntzen, O.J. (1998). Extending predictive capabilities to network models. SPE Journal. **December** 1998, 324-336.
- Orlov, S.I. (1970). Calculation and designing of coaxial resonators (in Russian). The Soviet Wireless, Moscow, USSR.
- Oliver, J., Guitart, J., Ayguadé, E., Navarro, N. and Torres, J. (2001). Strategies for the efficient exploitation of loop-level parallelism in Java. Concurrency and Computation: Practice and Experience. **13**(8-9), 663-680.
- Olson, J.F. and Rothman, D.H. (1997). Two-fluid flow in sedimentary rock: simulation, transport, and complexity. J. Fluid Mech. **341**, 343-370.
- Parker, J. (1989). Multiphase flow and transport in porous media. Rev. Geophy. **27**(3), 311-328.
- Peaceman, D.P. (1977). Fundamentals of numerical reservoir simulation: Developments in petroleum engineering, 6). Elsevier Scientific Publishing Company, Amsterdam, The Netherlands, 65-82.

- Peters, E.J. and Afzal, N. (1992). Characteristics of heterogeneities in permeable media with computed tomography imaging. J. Pet. Sci. Eng. **7**, 283-296.
- Peters, E.J. and Hardham, W.D. (1990). Visualization of fluid displacements in porous media using computed tomography imaging. J. Pet. Sci. Eng. **4**, 155-168.
- Peters, E.J. and Khataniar, S. (1987). The effect of instability on relative permeability curves obtained by dynamic-displacement method. SPE Formation Evaluation. **December** 1987, 469-474.
- Peters, E.J. and Flock, D.L., (1981). The onset of instability during two-phase immiscible displacement in porous media. SPE Journal. **21**(2), 249-258.
- Peters, E.J. (1979). Stability theory and viscous fingering in porous media. Ph.D. Thesis, University of Alberta, Canada.
- Polikar, M., Puttagunta, V.R., DeCastro, V and Ali, S.M.F. (1989). Relative permeability curves for bitumen and water in oilsand systems. JCPT. January-February 1989, **28**(1), 93-99.
- Philip, J. R. (1970). Flow in porous media. Annual Review of Fluid Mechanics. **2**, 177-203.
- Philip, J.R. (1973). Flow in porous media. In Theoretical & Applied Mechanics. Proc. 13th Int. Congr. Theor. & Appl. Mech., Moscow, 1972. 279-294. Springer-Verlag, Berlin, Germany.
- Philippsen, M., Boisvert, R. F., Getov, V.S., Pozo, R., Moreira, J., Gannon, D. and Geoffrey, C.F. (2001). PARA 2000, lecture notes in computer science (LNCS) 1947, 20-26. Springer-Verlag, Berlin.
- Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1992). Numerical recipe: The art of scientific computing in C. Cambridge University Press, Cambridge, UK. 50-51, 379-385.
- Purcell, W.R. (1949). Capillary pressures - their measurement using mercury and the calculation of permeability therefrom. Trans. AIME. **186**, 39-48.
- Qian, Y.H. (1990). Gaz sur réseau at théorie cinétique sur réseaux appliquée à l' équation de Navier-Stokes. Ph.D. thesis, Université Paris 6.
- Qian, Y.H., D'Humiéres, D. and Lallemand, P. (1992). Lattice BGK models for Navier-Stokes equation. Europhys. Lett. **17**, 479-484.
- Rakotomalala, N., Salin, D. and Yortsos, C.Y. (1995). Viscous coupling in a model porous medium geometry: Effect of fluid contact area. App. Scientific Res. **55**, 155-169.
- Rapport, L.A. and Leas, W.J. (1953). Properties of linear waterfloods. Trans. AIME.**198**.139-148.
- Rice, J.R. (1983). Numerical methods, software and analysis. McGraw-Hill Book Company Inc., New York, US. 320, 338.
- Richmond P.C. and Watson A.T. (1990). Estimation of multiphase flow functions from displacement experiments. SPE Reservoir Engineering. February 1990, 121-127.
- Rose, W. (2000). Myths about later-day extension of Darcy's law. J. Petrol. Sci. Eng. **26**, 187-198.
- Rose, W. (1999). Relative permeability ideas - Then and now (Richards to Leverett to Yuster, and Beyond). SPE paper number 57442 presented at the 1999 SPE regional meeting, Charleston, West Virginia, Oct. 1999.

- Rose, W., (1993). Coupling coefficients for two-phase flow in pore spaces of simple geometry. Transport in Porous Media. **10**, 293-296.
- Rose, W. (1990a). Coupling coefficients for two-phase flow in pore spaces of simple geometry. Transport in Porous Media **5**(1), 92-102.
- Rose, W., (1990b). Lagrangian simulation of coupled two-phase flows", Mathematical Geology. **22**(6), 641-654.
- Rose, W. (1988). Measuring transport coefficients necessary for the description of coupled two-phase flow of immiscible fluids in porous media. Transport in Porous Media, **3**(2), 163-171.
- Rothman, D.H., (1990). Macroscopic laws for immiscible two-phase flow in porous media: Results from numerical experiments. J. Geophysical Res. **95**(B6), 8663-8674.
- Rothman, D.H., (1988). Cellular- automation fluids: A model for flow in porous media. Geophysics. **53**(4), 509-518.
- Rothman, D.H. and Keller, J.M. (1988). Immiscible cellular-automation fluids. J. Stat. Phys., **52**(3/4), 1119-1127.
- Saeedi, J, (1979). A comparison of theoretical and experimental saturation profiles in porous media. A thesis submitted to the Faculty of Graduate Studies and Research of the University of Alberta, in partial fulfillment of the requirements for the degree of Master of Science in Petroleum Engineering.
- Sahimi, M. (1995). Flow and transport in porous media and fractured rock: From classical methods to modern approaches. VCH Verlagsgesellschaft mbH, Weinheim, Federal Republic of Germany.
- Santos, L.O.E., Philippi, P.C., Damiani, M.C. and Fernandes, C.P. (2002). Using three-dimensional reconstructed microstructures for predicting intrinsic permeability of reservoir rocks based on a Boolean lattice gas method. J. Petrol. Sci. **Eng**. 35, 109-124.
- Saraf, D.N. and McCaffery, F.G. (1982). Two and three-phase relative permeability measurement: A review. Petroleum recovery institute report number 81-8, Calgary, Alberta, Edmonton.
- Sarma, H.K. (1988). A study of the impact of instability on the immiscible displacement of one fluid by another. A thesis submitted to the Department of Mining, Metallurgical and Petroleum Engineering of the University of Alberta in partial fulfillment for the award of the degree of Doctor of Philosophy in Petroleum Engineering.
- Sarma, H.K. and Bentsen, R.G. (1990). Further experimental validation of the external drive technique. JCPT. **July-August** 1990, **29**(4), 75-85.
- Sarma, H.K. and Bentsen, R.G. (1989a). A new method for estimating relative permeabilities from unstabilized displacement data. JCPT. **July-August** 1989, **28**(4), 118-128.
- Sarma, H.K. and Bentsen, R.G. (1989b). A study of the impact of instability on relative permeability and capillary pressure. J. Petrol. Sci. Eng. **2**, 311-330.
- Satofuka, N. and Nishioka, T. (1999). Parallelization of lattice Boltzmann method for incompressible flow computations. Computational Mechanics. **23**, 164-171.
- Scheidegger, A.E. (1974). The physics of flow through porous media. 3rd Edition. University of Toronto Press, Toronto, Canada.

- Scott, P.H. and Rose, W. (1953). An explanation of the Yuster Effect. JPT. **November** 1953, 19-20.
- Shen, C. and Ruth, D.W. (1996). Impact of inlet boundary conditions on the numerical simulation of one-dimensional coreflooding. JCPT. January 1996, **35**(1), 19-24.
- Shen, C. and Ruth, D.W. (1994a). Solutions to Bentsen's equation with finite-element method. J. Petrol. Sci. Eng. **11**, 165-179.
- Shen, C. and Ruth, D.W. (1994b). Steady state solutions to the Bentsen's equation. Transport in Porous Media. **16**, 105-123.
- Shen, K.L.P and Qing, T. (1994). A new method for calculating oil-water relative permeabilities with consideration of capillary pressure. Mechanics and Practice. **16**(2), 46-52.
- Sigmund, P.M. and McCaffery, F.G. (1979). An improved unsteady state-state procedure for determining the relative permeability characteristics of heterogeneous porous media. SPE Journal. **February** 1979, 15-28.
- Slattery, J.C., (1970). Two-phase flow through porous media. AIChE J. **16**(3), 345-352.
- Slattery, J.C., (1969). Single-phase flow through porous media. AIChE J. **15**(6), 866-872.
- Slobod, R. L., Chambers, A. and Prehn, W.L. (1952) Use of centrifuge for determining connate water, residual oil, and capillary pressure curves of small core samples. Trans. AIME. **192**, 127-134.
- Spanos, T.J.T., dela Cruz, V. and Hube, J. (1988). An analysis of the theoretical foundations of relative permeability curves, AOSTRA J. Res. **4**, 181-192.
- Sprunt, E.S., Desal, K.P., Coles, M.E., Davis R.M., and Muegge, E.L. (1991), CT-Scan-Monitored electrical resistivity measurements show problem achieving homogenous saturation. SPE paper number 21433 presented at the 1991 SPE Middle East oil show, Bahrain, 2-5 March 1991.
- Stone, H.L. (1970). Probabilistic model for estimating three-phase relative permeability. JPT. **214**.Trans. AIME **249**.
- Stone, H.L. (1973). Estimation of three-phase relative permeability and residual oil data. JCPT. October-December, **12**(4), 53-61.
- Su, Q., Feng Q. and Shang, Z. (2000). Electrical impedance variation with water saturation in rock. Geophysics. **65**(1), 68-75.
- Swanson, B.F., (1979). Visualizing pores and nonwetting phase in porous rock. JPT, **January** 1979, 10-18.
- Tao, T.M. and Watson, A.T., (1984). Accuracy of JBN estimates of relative permeability: Part I, error analysis. SPE Journal. **April** 1984, 209-214.
- Thiruvathukal, G.K. (2002). Java at Middle Age: Enabling Java for computational science. Computing in Science and Engineering (CiSE), **January/February** 2002, 74-75, 77-84.
- Thomas, G.W. (1982). Principles of hydrocarbon reservoir simulation. International Human Resources Development Corporation, Boston, US.
- Toth, J., Bodi, T., Szucs, P. and Civan F. (2002). Convenient formulae for determination of relative permeability from unsteady-state displacements in core J. Petrol. Sci. Eng. **36**, 33-44.
- Trapp, J.A., (1976). On the relationship between continuum mixture theory and integral averaged equations for immiscible fluids. Int. J. Eng. Sci. **14**, 991-998.

- Urkedal, H., Ebeltoft, E., Nordtvedt, J.E., Watson, A.T. (2000). A new design of steady state type experiments for simultaneous estimation of two-phase flow function. SPE Reservoir Engineering. **June** 2000, 230-238.
- Ursin, J. -R. (1992). Detection of fluid saturation levels in porous media using gamma-ray tomography. J. Pet. Sci. Eng. **7**, 297-308.
- van Genabeek, O. and Rothman, D.H., (1996). Macroscopic manifestations of microscopic flows through porous media: Phenomenology from simulation. Annu. Rev. Earth Planet. Sci. **24**, 63-87.
- van Kats, F.M. and Egberts, P.J.P. (1999). Simulation of three-phase displacement mechanisms using a 2D Lattice-Boltzmann model. Transport in Porous Media. **37**, 55-58.
- van Spronsen, E. (1982). Three-phase relative permeability measurements using the centrifuge method. SPE/DOE paper number 10688 presented at the 1982 SPE/DOE 3rd joint symposium on enhanced oil recovery, Tulsa, Oklahoma, April 4-7 1982.
- Vannier, H.J. and Ellington, W.A. (1988). Quantitative computed tomography. Rev. Progr. Quantitative Nondestructive Evaluation, 7a.
- Vinegar, H.J. and Wellington, S.L. (1987). Tomographic imaging of three-phase flow experiments. Rev. Sci. Instrum. **58**(1), 96-107.
- Virnovsky, G.A., Guo, Y., and Skjaeveland, S.M. (1995). Relative permeability and capillary pressure concurrently determined from steady state flow experiments. Proceedings, 8th European IOR symposium, Vienna, Austria, 1995.
- Wang, C.Y. (1997). An alternative description of viscous coupling in two-phase flow through porous media. Transport in porous media. **28**, 205-219.
- Wang, C.Y. and Beckermann, C. (1993). A two-phase mixture model of liquid-gas flow and heat transfer in capillary porous media: 1. Formulation. Int. J. Heat Transfer. **36**(11), 2747-2758.
- Wang, C.Y. and Cheng, P. (1996). A multiphase mixture model for multiphase, multicomponent transport in capillary porous media: I. Model development. Int. J. Heat Mass Transfer. **39**(17), 3607-3618.
- Welge, H.J. (1952). A simplified method for computing oil recovery by gas or water drive. Trans. AIME. **195**, 91.
- West, L.J., Handley, K., and Huang, Y. (2003). Radar frequency dielectric dispersion in sandstone: Implications for determination of moisture and clay content. Water Resources Research. **February** 2003. **39(2)**, 1:1-12.
- Whitaker, S. (1986). Flow in porous media II: The governing equations for immiscible, two-phase flow. Transport in Porous Media. **1**, 105-125.
- Whitaker, S., (1973). The transport equations for multi-phase systems. Chem. Eng. Sci. **28**, 139-147.
- Whitaker, S., (1967). Diffusion and dispersion in porous media. AIChE J. **May** 1967, 420-427.
- Wooding, R.A., and Morel-Seytoux, H.J. (1976). Multiphase fluid flow through porous media. Annual Review of Fluid Mechanics. **8**, 233-274.
- Wyckoff, R.D. and Botset, H.G. (1936). The flow of gas-liquid mixtures through unconsolidated sands. **September** 1936. Physics. **7**, 325-345.

- Yadav, G.D., Dullien, F.A.L., Chatzis, I. and McDonald, I.F. (1987). Microscopic distribution of wetting and nonwetting phases in sandstones during immiscible displacement. SPE Reservoir Engineering. **May** 1987, 137-147.
- Yuan, J. -Y., Coombe, D., Law, D.H.-S. and Babchin, A. (2001). Determination of the relative permeability matrix coefficients. Petroleum Society paper number 2001-002 presented at the Petroleum Society's Canadian international conference 2001, Calgary, Alberta, Canada, June 12 -14, 2001.
- Young, L.C. (1978). An efficient finite element method for reservoir simulation. SPE paper number 7413 presented at the 1976 53rd annual fall technical conference and exhibition of the SPE of AIME, Houston, Texas, Oct. 1-3, 1978.
- Yuster, S.T. (1951). Theoretical consideration of multiphase flow in idealized capillary systems. Proceeding of the Third World Petroleum Congress. **2**, 437-445.
- Zarcone, C. and Lenormand, R. (1994). Détermination expérimentale du couplage visqueux dans les écoulements diphasiques en milieu poreux, C. R. Acad. Sci., Paris, Series II, **318**, 1429-1438.
- Zhou, D., Jia, L., Kamath, J. and Kovscet, A.R. (2002). Scaling of counter-current imbibition processes in low-permeability porous media. J. Petrol. Sci. Eng. **33**, 61-74.

APPENDIX A1: Source Code Listings for the Interfacial Coupling Simulator

The ics package is made up of four source files (.java files) compiled into their byte codes equivalents files (.class files). These java files, briefly described in Section 5.12.5, were packaged into an executable jar (Java archive) file - ICS.jar. The jar file can be executed (run) by simply double clicking on it on any computer on which a JVM (Java virtual machine) is installed. The complete source codes (.java files) and their byte codes representations (.class files), as well as the executable jar (Java archive) file - ICS.jar, are contained in the CD-ROM at the back of this thesis. The source codes contained in all the java files are given in this Appendix A.

i) The Source Codes Listings for Ics.java

```
package ics;

import javax.swing.UIManager;
import java.awt.*;
import ics.*;

/**
 * <p>Title: Interfacial Coupling Simulator</p>
 * <p>Description: Program  for Testing Coupling</p>
 * <p>Copyright: Copyright (c) 2004</p>
 * <p>University: University of Alberta</p>
 * <p>Program: Petroleum Engineering</p>
 * <p>Degree: Doctor of Philosophy - Ph.D</p>
 * @author O.R. Ayodele
 * @version 1.0
 */

 public class Ics {

  boolean packFrame = false;

  //Construct the application
  public Ics() {
    IcsFrame frame = new IcsFrame();

    //Validate frames that have preset sizes
    //Pack frames that have useful preferred size info, e.g. from their layout
    if (packFrame) {
     frame.pack();
    }
    else {
     frame.validate();
    }

    //Center the window
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = frame.getSize();

    if (frameSize.height > screenSize.height) {
      frameSize.height = screenSize.height;
    }

    if (frameSize.width > screenSize.width) {
     frameSize.width = screenSize.width;
    }

    frame.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);
    frame.setVisible(true);
  }

  //Main method
  public static void main(String[] args) {
    new Ics();
  }
}
```

ii) The Source Codes Listings for IcsFrame.java

```
package ics;

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;
import javax.swing.border.*;
import ics.*;

public class IcsFrame extends JFrame {
 //Panel
 JPanel contentPane;

 //Menu bar and menu items
 JMenuBar jMenuBar1 = new JMenuBar();
 JMenu jMenuFile = new JMenu();
 JMenuItem jMenuFileNew = new JMenuItem();
 JMenuItem jMenuFileExit = new JMenuItem();
 JMenu jMenuHelp = new JMenu();
 JMenuItem jMenuHelpAbout = new JMenuItem();

 //Buttons
 JButton jButton1 = new JButton();
 JButton jButton2 = new JButton();
 JButton jButton3 = new JButton();

 //Image Icons
 ImageIcon image1;
 ImageIcon image2;
 ImageIcon image3;
 JMenuItem jMenuNew = new JMenuItem();

 //Border
 Border border1;

 //Grid layout
 GridLayout gridLayout1 = new GridLayout();

 //A static variable to count the number of new "inputbox"
 static int IcsClassCount = 0;

 //Construct the frame
 public IcsFrame() {
  try {
   jbInit();
  }

  catch(Exception e) {
   e.printStackTrace();
  }
 }

 //Component initialization
 private void jbInit() throws Exception {
  contentPane = (JPanel) this.getContentPane();
  border1 = BorderFactory.createEtchedBorder(Color.white,new Color(148, 145, 140));
  contentPane.setLayout(gridLayout1);
  this.setSize(new Dimension(900, 700));
  this.setTitle("Interfacial Coupling Simulator");
  jMenuFile.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuFile.setMnemonic('F');
  jMenuFile.setText("File");
  jMenuFileNew.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuFileNew.setMnemonic('N');
  jMenuFileNew.setText("New");

  jMenuFileNew.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent e) {
    jMenuFileNew_actionPerformed(e);
   }
  });

  jMenuFileExit.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuFileExit.setMnemonic('E');
  jMenuFileExit.setText("Exit");

  jMenuFileExit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      jMenuFileExit_actionPerformed(e);
    }
  });

  jMenuHelp.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuHelp.setMnemonic('H');
  jMenuHelp.setText("Help");
  jMenuHelpAbout.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuHelpAbout.setMnemonic('A');
  jMenuHelpAbout.setText("About");

  jMenuHelpAbout.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
      jMenuHelpAbout_actionPerformed(e);
    }
  });

  jMenuNew.setFont(new java.awt.Font("Dialog", 1, 12));
  jMenuNew.setMnemonic('N');
  jMenuFile.addSeparator();
  jButton1.setBorder(border1);
  contentPane.setEnabled(true);
  jMenuFile.add(jMenuFileNew);
  jMenuFile.addSeparator();
  jMenuFile.add(jMenuFileExit);
  jMenuHelp.addSeparator();
  jMenuHelp.add(jMenuHelpAbout);
  jMenuBar1.add(jMenuFile);
  jMenuBar1.add(jMenuHelp);
  this.setJMenuBar(jMenuBar1);
  this.toBack();
}

  //File | New action performed
  public void jMenuFileNew_actionPerformed(ActionEvent e) {
    JDesktopPane n = new JDesktopPane();
    this.getContentPane().add(n);

    //An instance of IcsClass
    IcsClass inputbox = new IcsClass();

    //Update the static variable by adding 1.
    IcsClassCount = IcsClassCount + 1;

    //Set inputbox size and display
    Dimension dlgSize = getPreferredSize();
    Dimension frmSize = getSize();
    Point loc = getLocation();
    inputbox.jif.setLocation((frmSize.width - dlgSize.width) / 2 + loc.x, (frmSize.height - dlgSize.height) / 2 + loc.y);
    inputbox.jif.setResizable(false);
    inputbox.jif.setBounds(40, 150, 825, 430);
    inputbox.jif.setTitle("Simulation Input Settings " + IcsClassCount);
    inputbox.jif.show();
    n.add(inputbox.jif);
    inputbox.jif.setLocation(20, 213);

    inputbox.f1.setLocation(20, 1);
    inputbox.f1.setResizable(false);
    inputbox.f1.setSize(inputbox.fv1.width, inputbox.fv1.height);
    inputbox.f1.setVisible(true);

    inputbox.f2.setLocation(250, 1);
    inputbox.f2.setResizable(false);
    inputbox.f2.setSize(inputbox.fv2.width, inputbox.fv2.height);
    inputbox.f2.setVisible(true);
    n.add(inputbox.f2);
    n.add(inputbox.f1);


    //Progress bar code for monitoring simuation run status
    inputbox.jProgressBar1.setIndeterminate(false);

    //Reset all of these values to zeros and initial title
    inputbox.flowCount = 0;
    inputbox.residualValueSum = 0;
    inputbox.timeDivision = 0;
    inputbox.flowIterationNumber = 0;
     for(int i = 0; i < inputbox.bNum; i++){
      inputbox.copyE[i] = 0;
    }

  }

  //File | Exit action performed
  public void jMenuFileExit_actionPerformed(ActionEvent e) {
    System.exit(0);
  }

  //Help | About action performed
  public void jMenuHelpAbout_actionPerformed(ActionEvent e) {
    IcsFrame_AboutBox dlg = new IcsFrame_AboutBox(this);
    Dimension dlgSize = dlg.getPreferredSize();
    Dimension frmSize = getSize();
    Point loc = getLocation();
    dlg.setLocation((frmSize.width - dlgSize.width) / 2 + loc.x, (frmSize.height - dlgSize.height) / 2 + loc.y);
    dlg.setModal(true);
    dlg.show();
  }

  //Overridden so we can exit when window is closed
  protected void processWindowEvent(WindowEvent e) {
    super.processWindowEvent(e);

    if (e.getID() == WindowEvent.WINDOW_CLOSING) {
      jMenuFileExit_actionPerformed(null);
    }
  }
}
```

iii) The Source Codes Listings for IcsClass.java

```
package ics;

import java.awt.*;
import java.awt.event.*;
import java.awt.FileDialog.*;
import java.io.*;
import java.io.Reader.*;
import java.text.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.border.*;
import ics.*;

/**  IcsClass (interfacial coupling class) for solving Bentsen's Equation
 *   which incorporates interfacial coupling using the fully-implicit finite
 *   difference method. The solution of the resulting systems of linear
 *   equations is based on the "ThomasAlgorithmSolution" and "newtonJacobi"
 *   methods, which are objects in the IcsClass class.
 *
 * @author Oluropo Rufus Ayodele
 * @version 1.0
 */

public class IcsClass {
```

```
/**
 * Initiation of Components (Constructor)
 */
 public IcsClass() {
  try {
    initiate();
  }
  catch(Exception e) {
    e.printStackTrace();
  }
}

//CONSTRUCTION OF GUI STARTS HERE

 //Declaration of all public and common GUI variables (fields) starts here
 JInternalFrame jif = new JInternalFrame();
 JInternalFrame  distanceOption = new JInternalFrame();
 JInternalFrame  relativeEffective = new JInternalFrame();
 JProgressBar jProgressBar1 = new JProgressBar();
 String updateTitle;
 String updateTitleNow;
 DecimalFormat dataFormatUpdateTime2 =  new DecimalFormat("0.00");
 DecimalFormat dataFormatUpdateTime8 =  new DecimalFormat("0.00000000");
 DecimalFormat dataFormat6 =  new DecimalFormat("0.000000");
 JInternalFrame f1 = new JInternalFrame("Fractional Flow Profile");
 JInternalFrame f2 = new JInternalFrame("Saturation Profile");
 FlowView fv1 = new FlowView();
 FlowView fv2 = new FlowView();
 private JPanel jPanel1 = new JPanel();
 private JPanel jPanel2 = new JPanel();
 private JPanel jPanel3 = new JPanel();
 private JPanel jPanel4 = new JPanel();
 private Border border1;
 private TitledBorder titledBorder1;
 private Border border2;
 private TitledBorder titledBorder2;
 private Border border3;
 private TitledBorder titledBorder3;
 private Border border4;
 private TitledBorder titledBorder4;
 private JButton runSimulation = new JButton();
 private Border border5;
 private TitledBorder titledBorder5;
 private JTextField jTextField1 = new JTextField();
 private JTextField jTextField2 = new JTextField();
 private JTextField jTextField3 = new JTextField();
 private JLabel jLabel1 = new JLabel();
 private JTextField jTextField4 = new JTextField();
 private JLabel jLabel2 = new JLabel();
 private JLabel jLabel3 = new JLabel();
 private JLabel jLabel4 = new JLabel();
 private JLabel jLabel5 = new JLabel();
 private JLabel jLabel6 = new JLabel();
 private JLabel jLabel7 = new JLabel();
 private JTextField jTextField5 = new JTextField();
 private JTextField jTextField6 = new JTextField();
 private JTextField jTextField7 = new JTextField();
 private JTextField jTextField13 = new JTextField();
 private JTextField jTextField12 = new JTextField();
 private JTextField jTextField11 = new JTextField();
 private JTextField jTextField10 = new JTextField();
 private JTextField jTextField9 = new JTextField();
 private JTextField jTextField8 = new JTextField();
 private JLabel jLabel8 = new JLabel();
 private JLabel jLabel9 = new JLabel();
 private JLabel jLabel10 = new JLabel();
 private JLabel jLabel11 = new JLabel();
 private JLabel jLabel12 = new JLabel();
 private JLabel jLabel13 = new JLabel();
 private JLabel jLabel14 = new JLabel();
 private JTextField jTextField14 = new JTextField();
 private JTextField jTextField15 = new JTextField();
 private JTextField jTextField16 = new JTextField();
 private JLabel jLabel15 = new JLabel();
 private JTextField jTextField17 = new JTextField();
 private JLabel jLabel16 = new JLabel();
 private JTextField jTextField18 = new JTextField();
 private JLabel jLabel17 = new JLabel();
 private JTextField jTextField19 = new JTextField();
 private JLabel jLabel18 = new JLabel();
 private JLabel jLabel19 = new JLabel();
 private JLabel jLabel20 = new JLabel();
 private JLabel jLabel21 = new JLabel();
 private JTextField jTextField20 = new JTextField();
 private JTextField jTextField21 = new JTextField();
 private JButton jButton5 = new JButton();
 private JButton stopSimulation = new JButton();
 private JPanel jPanel6 = new JPanel();
 private JPanel jPanel7 = new JPanel();
 private JTextField jTextField22 = new JTextField();
 private JLabel jLabel22 = new JLabel();
 private JTextField jTextField23 = new JTextField();
 private JLabel jLabel23 = new JLabel();
 private TitledBorder titledBorder6;
 private TitledBorder titledBorder7;
 private JSlider jSlider1 = new JSlider();
 private JPanel jPanel9 = new JPanel();
 private Border border6;
 private Border border7;
 private TitledBorder titledBorder8;
 private JPanel jPanel8 = new JPanel();
 private Border border8;
 private TitledBorder titledBorder9;
 private TitledBorder titledBorder92;
 private JButton clearAll = new JButton();
 private Border border9;
 private TitledBorder titledBorder10;
 private ButtonGroup R12 = new ButtonGroup();
 private ButtonGroup Flow = new ButtonGroup();
 private ButtonGroup Coupling = new ButtonGroup();
 private ButtonGroup SolutionMethod = new ButtonGroup();
 private ButtonGroup Solver = new ButtonGroup();
 private JRadioButton r12 = new JRadioButton();
 private JRadioButton jRadioButton2 = new JRadioButton();
 private Border border10;
 private TitledBorder titledBorder11;
 private Border border11;
 private Border border12;
 private TitledBorder titledBorder12;
 private JTextField jTextField24 = new JTextField();
 private JTextField jTextField25 = new JTextField();
 private JLabel jLabel24 = new JLabel();
 private JLabel jLabel25 = new JLabel();
 private Border border13;
 private TitledBorder titledBorder13;
 private JRadioButton viscousCoupling = new JRadioButton();
 private JRadioButton capillaryCoupling = new JRadioButton();
 private JRadioButton bothCouplings = new JRadioButton();
 private JRadioButton noCoupling = new JRadioButton();
 private JPanel jPanel5 = new JPanel();
 private Border border14;
 private TitledBorder titledBorder14;
 private JTextField jTextField26 = new JTextField();
 private JTextField jTextField27 = new JTextField();
 private JTextField jTextField28 = new JTextField();
 private Border border15;
 private TitledBorder titledBorder15;
 private JLabel jLabel28 = new JLabel();
 private JLabel jLabel29 = new JLabel();
 private JLabel jLabel210 = new JLabel();
 private Border border16;
 private TitledBorder titledBorder16;
 private JPanel jPanel10 = new JPanel();
 private Border border17;
 private TitledBorder titledBorder17;
 private JTextField jTextField30 = new JTextField();
 private JTextField jTextField31 = new JTextField();
 private JTextField jTextField32 = new JTextField();
 private JTextField jTextField33 = new JTextField();
 private JLabel jLabel27 = new JLabel();
 private JLabel jLabel211 = new JLabel();
 private JLabel jLabel212 = new JLabel();
 private JLabel jLabel213 = new JLabel();
 private Border border18;
 private JLabel jLabel214 = new JLabel();
 private JTextField jTextField37 = new JTextField();
 private JLabel jLabel215 = new JLabel();
 private JLabel jLabel216 = new JLabel();
 private JTextField jTextField36 = new JTextField();
 private JTextField jTextField35 = new JTextField();
 private JTextField jTextField34 = new JTextField();
 private JLabel jLabel217 = new JLabel();
 private JPanel jPanel13 = new JPanel();
 private Border border19;
 private TitledBorder titledBorder18;
 private Border border20;
 private Border border21;
 private TitledBorder titledBorder19;
 private JTextField jTextField41 = new JTextField();
 private JLabel jLabel219 = new JLabel();
 private JTextField jTextField40 = new JTextField();
 private JTextField jTextField39 = new JTextField();
 private JTextField jTextField38 = new JTextField();
 private Border border22;
 private TitledBorder titledBorder20;
 private JTextField jTextField29 = new JTextField();
 private Border border23;
 private TitledBorder titledBorder21;
 private JLabel jLabel2112 = new JLabel();
 private Border border24;
 private TitledBorder titledBorder22;
 private JButton saveInputData = new JButton();
 private JButton openInputData = new JButton();
 private Border border25;
 private TitledBorder titledBorder23;
 private Border border26;
 private TitledBorder titledBorder24;
 private Border border27;
 private JLabel jLabel220 = new JLabel();
 private Border border28;
 private JLabel jLabel2110 = new JLabel();
 private JLabel jLabel2111 = new JLabel();
 private JLabel jLabel2113 = new JLabel();
 private JPanel jPanel14 = new JPanel();
 private JPanel jPanel11 = new JPanel();
 private Border border29;
 private TitledBorder titledBorder25;
 private Border border30;
 private TitledBorder titledBorder26;
 private Border border31;
 private TitledBorder titledBorder27;
 private Border border32;
 private JRadioButton implicitNewtonRaphson = new JRadioButton();
 private JRadioButton implicitNewtonJacobi = new JRadioButton();
 private JRadioButton no = new JRadioButton();
 private JRadioButton yes = new JRadioButton();
 private Border border33;
 private TitledBorder titledBorder28;
 private JPanel jPanel15 = new JPanel();
 private Border border34;
 private TitledBorder titledBorder29;
 private Border border35;
 private TitledBorder titledBorder30;
 private Border border36;
 private TitledBorder titledBorder31;
 private JPanel jPanel16 = new JPanel();
 private Border border37;
 private JScrollPane jScrollPane1 = new JScrollPane();
 boolean isStop = true;
```

```
//Declaration of all public and common GUI variables (fields) ends here*/

//Initiation of GUI Starts here
private void initiate() throws Exception {
  border1 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder1 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Fluid Properties");
  border2 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder2 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Reservoir
Properties");
  border3 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder3 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Simulation
Settings");
  border4 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder4 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Saturation Block
(Grid)");
  titledBorder6 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Hydrodynamic
Effect (R12)");
  border5 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder5 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"");
  border6 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  border7 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder8 = new TitledBorder(new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134)),"Coupling
Options");
  border8 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder9 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Pc Coeff");
  border9 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder10 = new TitledBorder(border9,"Events");
  border10 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder11 = new TitledBorder(border10,"Events");
  border11 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Event");
  border12 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder12 = new TitledBorder(border12,"Events");
  border13 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  border14 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder14 = new TitledBorder(BorderFactory.createEmptyBorder(),"Event");
  border15 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder15 = new TitledBorder(border15,"Viscous Coupling Term");
  border16 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  border17 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder17 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Boundary
Condition (S*) a = 1.0");
  border18 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Ko Coeff");
  border19 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder18 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Krw Coeff");
  border20 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Krw Coeff");
  border21 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder19 = new TitledBorder(border21,"Ko Coeff");
  border22 = BorderFactory.createEmptyBorder();
  titledBorder20 = new TitledBorder(new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134)),"Kro
Coeff");
  border23 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder21 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Events");
  border24 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder22 = new TitledBorder(border24,"Viscous Coupling Term");
  border25 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder23 = new TitledBorder(border25,"Events");
  border26 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder24 = new TitledBorder(border26,"Events");
  border27 = new EtchedBorder(EtchedBorder.RAISED,new Color(197, 187, 178),new Color(96, 91, 87));
  border28 = BorderFactory.createEmptyBorder();
  border29 = BorderFactory.createEmptyBorder();
  titledBorder25 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Initial
Conditions");
  border30 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
  titledBorder26 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Coupling");
  border31 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  border32 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Interactive");
  border33 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  titledBorder28 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Solver ");
  border34 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  border35 = BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134));
  border36 = new EtchedBorder(EtchedBorder.RAISED,Color.white,new Color(134, 134, 134));
```

```
border37 = new TitledBorder(BorderFactory.createEtchedBorder(Color.white,new Color(134, 134, 134)),"Viscous Coupling ");
jif.getContentPane().setLayout(null);
jPanel1.setBorder(titledBorder1);
jPanel1.setBounds(new Rectangle(5, 2, 372, 236));
jPanel1.setLayout(null);
jPanel2.setLayout(null);
jPanel2.setBounds(new Rectangle(379, 3, 221, 236));
jPanel2.setBorder(titledBorder2);
jPanel3.setLayout(null);
jPanel3.setBounds(new Rectangle(4, 233, 202, 163));
jPanel3.setBorder(titledBorder3);
jPanel4.setLayout(null);
jPanel4.setBorder(titledBorder25);
jPanel4.setBounds(new Rectangle(207, 309, 166, 89));
jif.setFont(new java.awt.Font("Dialog", 1, 12));
jif.setResizable(true);
jif.setTitle("    Simulator Input Section");
runSimulation.setText("Run Simulation");
runSimulation.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    runSimulation_actionPerformed(e);
  }
});
runSimulation.setBorder(BorderFactory.createEtchedBorder());
runSimulation.setMnemonic('R');
runSimulation.setBounds(new Rectangle(662, 340, 124, 17));
jTextField1.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField1.setText("812.3");
jTextField1.setBounds(new Rectangle(127, 14, 57, 24));
jTextField2.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField2.setText("998.2");
jTextField2.setBounds(new Rectangle(127, 44, 57, 24));
jTextField3.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField3.setText("0.04762");
jTextField3.setBounds(new Rectangle(127, 75, 57, 24));
jLabel1.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel1.setText("Oil Density (Kg/m^3)");
jLabel1.setBounds(new Rectangle(10, 16, 108, 29));
jTextField4.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField4.setText("0.0010");
jTextField4.setBounds(new Rectangle(127, 105, 57, 24));
jLabel2.setBounds(new Rectangle(10, 47, 114, 29));
jLabel2.setText("Water Density (Kg/m^3)");
jLabel2.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel3.setBounds(new Rectangle(10, 77, 108, 29));
jLabel3.setText("Oil Viscosity (Pa.s)");
jLabel3.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel4.setBounds(new Rectangle(10, 108, 108, 29));
jLabel4.setText("Water Viscosity (Pa.s)");
jLabel4.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel5.setBounds(new Rectangle(10, 139, 108, 29));
jLabel5.setText("Injection Time (sec)");
jLabel5.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel6.setBounds(new Rectangle(10, 169, 112, 29));
jLabel6.setText("Water Inj. Rate (m^3/s)");
jLabel6.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel7.setBounds(new Rectangle(10, 200, 126, 29));
jLabel7.setText("Flow No. of Time Step");
jLabel7.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField5.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField5.setText("360");
jTextField5.setBounds(new Rectangle(127, 135, 57, 24));
jTextField6.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField6.setText("9.0E-9");
jTextField6.setBounds(new Rectangle(127, 166, 57, 24));
jTextField7.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField7.setText("5");
jTextField7.setBounds(new Rectangle(127, 196, 57, 24));
jTextField13.setBounds(new Rectangle(308, 167, 57, 24));
jTextField13.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField13.setText("0.00001");
jTextField12.setBounds(new Rectangle(308, 136, 57, 24));
jTextField12.setBorder(BorderFactory.createLineBorder(Color.black));
```

```
jTextField12.setText("50");
jTextField11.setBounds(new Rectangle(308, 106, 57, 24));
jTextField11.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField11.setText("0.2000");
jTextField10.setBounds(new Rectangle(308, 76, 57, 24));
jTextField10.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField10.setText("0.1600");
jTextField9.setBounds(new Rectangle(308, 45, 57, 24));
jTextField9.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField9.setText("9.9E-12");
jTextField8.setBounds(new Rectangle(308, 15, 57, 24));
jTextField8.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField8.setText("9.0E-12");
jLabel8.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel8.setText("Length (m)");
jLabel8.setBounds(new Rectangle(8, 16, 126, 29));
jLabel9.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel9.setText("Flow Covg. Tolerance");
jLabel9.setBounds(new Rectangle(191, 170, 112, 29));
jLabel10.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel10.setText("Flow Count Tolerance");
jLabel10.setBounds(new Rectangle(191, 140, 108, 29));
jLabel11.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel11.setText("Sor (fraction)");
jLabel11.setBounds(new Rectangle(191, 109, 108, 29));
jLabel12.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel12.setText("Swi (fraction)");
jLabel12.setBounds(new Rectangle(191, 78, 108, 29));
jLabel13.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel13.setText("Eff.K to Wat. (m^2)");
jLabel13.setBounds(new Rectangle(191, 48, 122, 29));
jLabel14.setBounds(new Rectangle(191, 17, 117, 29));
jLabel14.setText("Eff.K to Oil (m^2)");
jLabel14.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel14.setToolTipText("");
jTextField14.setBounds(new Rectangle(151, 21, 57, 16));
jTextField14.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField14.setText("0.60");
jTextField15.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField15.setText("0.10");
jTextField15.setBounds(new Rectangle(151, 47, 57, 16));
jTextField16.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField16.setText("0.09");
jTextField16.setBounds(new Rectangle(151, 72, 57, 16));
jLabel15.setToolTipText("");
jLabel15.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel15.setText("Thickness (m)");
jLabel15.setBounds(new Rectangle(10, 42, 117, 29));
jTextField17.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField17.setText("395.00");
jTextField17.setBounds(new Rectangle(151, 98, 57, 16));
jLabel16.setBounds(new Rectangle(10, 67, 117, 29));
jLabel16.setText("Height/Elevation (m)");
jLabel16.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField18.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField18.setText("0.35");
jTextField18.setBounds(new Rectangle(151, 123, 57, 16));
jLabel17.setBounds(new Rectangle(10, 93, 117, 29));
jLabel17.setText("Pore Volume (m^3)");
jLabel17.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField19.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField19.setText("90.00");
jTextField19.setBounds(new Rectangle(151, 149, 57, 16));
jLabel18.setBounds(new Rectangle(10, 119, 117, 29));
jLabel18.setText("Porosity (fraction)");
jLabel18.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel19.setBounds(new Rectangle(10, 145, 117, 29));
jLabel19.setText("Inclination (Degrees)");
jLabel19.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel20.setBounds(new Rectangle(10, 170, 145, 29));
jLabel20.setText("Displacement Pressure (Pa)");
jLabel20.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel21.setBounds(new Rectangle(10, 196, 136, 29));
```

```
jLabel21.setText("Ac, Area of Pc Curve (Pa)");
jLabel21.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField20.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField20.setText("500.00");
jTextField20.setBounds(new Rectangle(151, 174, 57, 16));
jTextField21.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField21.setToolTipText("");
jTextField21.setText("1200.00");
jTextField21.setBounds(new Rectangle(151, 200, 57, 16));
jButton5.setBounds(new Rectangle(205, 197, 159, 24));
jButton5.setBorder(BorderFactory.createEtchedBorder());
jButton5.setMnemonic('C');
jButton5.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    jButton5_actionPerformed(e);
  }
});
jButton5.setText("Clear Fluid Properties");
stopSimulation.setBounds(new Rectangle(662, 320, 124, 17));
stopSimulation.setBorder(BorderFactory.createEtchedBorder());
stopSimulation.setToolTipText("");
stopSimulation.setMnemonic('U');
stopSimulation.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    stopSimulation_actionPerformed(e);
  }
});
stopSimulation.setText("Stop Simulation");
jPanel7.setLayout(null);
jPanel7.setBounds(new Rectangle(4, 79, 187, 76));
jPanel7.setBorder(titledBorder4);
jTextField22.setBounds(new Rectangle(115, 37, 51, 18));
jTextField22.setText("0.00");
jTextField22.disable();
jTextField22.setBorder(BorderFactory.createLineBorder(Color.black));
jLabel22.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel22.setText("Value of a for  R12");
jLabel22.setBounds(new Rectangle(17, 31, 94, 29));
jTextField23.setEnabled(false);
jTextField23.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField23.setText("10");
jTextField23.setBounds(new Rectangle(116, 53, 57, 16));
jTextField23.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    jTextField23_actionPerformed(e);
  }
});
jLabel23.setBounds(new Rectangle(16, 48, 69, 29));
jLabel23.setText("Block Number");
jLabel23.setFont(new java.awt.Font("Dialog", 1, 10));
jSlider1.setMajorTickSpacing(5);
jSlider1.setMinimum(10);
jSlider1.setMaximum(50);
jSlider1.setMinorTickSpacing(1);
jSlider1.setPaintTicks(true);
jSlider1.setValue(10);
jSlider1.setBorder(BorderFactory.createEtchedBorder());
jSlider1.setBounds(new Rectangle(15, 18, 159, 31));
jSlider1.addChangeListener(new ChangeListener() {
  public void stateChanged(ChangeEvent e) {
    jSlider1_actionChange(e);
  }
  });
jPanel9.setBorder(border32);
jPanel9.setBounds(new Rectangle(207, 234, 80, 68));
jPanel9.setLayout(null);
no.setBounds(new Rectangle(10, 22, 65, 12));
no.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
   no_actionPerformed(e);
  }
});
no.setText("No");
no.setMnemonic('S');
no.setBorder(null);
no.setFont(new java.awt.Font("Dialog", 1, 10));
yes.setBounds(new Rectangle(10, 45, 65, 12));
yes.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    yes_actionPerformed(e);
  }
});
yes.setText("Yes");
yes.setMnemonic('U');
yes.setSelected(true);
yes.setBorder(null);
yes.setFont(new java.awt.Font("Dialog", 1, 10));
jPanel8.setBorder(titledBorder9);
jPanel8.setBounds(new Rectangle(609, 128, 110, 113));
clearAll.setBounds(new Rectangle(662, 301, 124, 17));
clearAll.setBorder(BorderFactory.createEtchedBorder());
clearAll.setMnemonic('L');
clearAll.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    clearAll_actionPerformed(e);
  }
});
clearAll.setText("Clear All");
r12.setFont(new java.awt.Font("Dialog", 1, 10));
r12.setBorder(null);
r12.setMnemonic('C');
r12.setText("Compute R12");
r12.setBounds(new Rectangle(12, 21, 86, 13));
r12.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    r12_actionPerformed(e);
  }
});
jRadioButton2.setBounds(new Rectangle(113, 21, 61, 13));
jRadioButton2.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    jRadioButton2_actionPerformed(e);
  }
});
jRadioButton2.setText(" R12=1");
jRadioButton2.setMnemonic('R');
jRadioButton2.setSelected(true);
jRadioButton2.setBorder(null);
jRadioButton2.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField24.setBounds(new Rectangle(95, 24, 57, 20));
jTextField24.setText("0.0");
jTextField24.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField25.setBounds(new Rectangle(95, 56, 57, 20));
jTextField25.setText("1.0");
jTextField25.setBorder(BorderFactory.createLineBorder(Color.black));
jLabel24.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel24.setText("Distance (m)");
jLabel24.setBounds(new Rectangle(9, 19, 69, 29));
jLabel25.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel25.setText("fw (fraction)");
jLabel25.setBounds(new Rectangle(9, 50, 69, 29));
viscousCoupling.setFont(new java.awt.Font("Dialog", 1, 10));
viscousCoupling.setBorder(null);
viscousCoupling.setMnemonic('0');
viscousCoupling.setSelected(true);
viscousCoupling.setText("Viscous");
viscousCoupling.setBounds(new Rectangle(743, 147, 65, 23));
viscousCoupling.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    viscousCoupling_actionPerformed(e);
  }
});
capillaryCoupling.setFont(new java.awt.Font("Dialog", 1, 10));
capillaryCoupling.setBorder(null);
capillaryCoupling.setMnemonic('0');
capillaryCoupling.setText("Capillary");
capillaryCoupling.setBounds(new Rectangle(743, 168, 65, 23));
capillaryCoupling.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    capillaryCoupling_actionPerformed(e);
  }
});
bothCouplings.setFont(new java.awt.Font("Dialog", 1, 10));
bothCouplings.setBorder(null);
bothCouplings.setActionCommand("bothCouplings");
bothCouplings.setMnemonic('C');
bothCouplings.setText("Both");
bothCouplings.setBounds(new Rectangle(743, 189, 65, 23));
bothCouplings.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    bothCouplings_actionPerformed(e);
  }
});
noCoupling.setFont(new java.awt.Font("Dialog", 1, 10));
noCoupling.setBorder(null);
noCoupling.setMnemonic('C');
noCoupling.setText("None");
noCoupling.setBounds(new Rectangle(743, 210, 65, 23));
noCoupling.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    noCoupling_actionPerformed(e);
  }
});
jPanel5.setBorder(titledBorder26);
jPanel5.setBounds(new Rectangle(723, 127, 94, 113));
jTextField26.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField26.setText("0.05");
jTextField26.setBounds(new Rectangle(521, 266, 97, 20));
jTextField27.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField27.setText("1.00");
jTextField27.setBounds(new Rectangle(521, 290, 97, 20));
jTextField28.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField28.setText("0.30");
jTextField28.setBounds(new Rectangle(521, 313, 97, 20));
jLabel28.setBounds(new Rectangle(401, 259, 117, 29));
jLabel28.setText("Coefficient b");
jLabel28.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel29.setBounds(new Rectangle(401, 285, 120, 29));
jLabel29.setText("Coefficient c");
jLabel29.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel210.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel210.setText("Coefficient d or S*");
jLabel210.setBounds(new Rectangle(401, 312, 124, 29));
jPanel10.setBorder(titledBorder17);
jPanel10.setBounds(new Rectangle(382, 241, 246, 100));
jTextField30.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField30.setText("0.00");
jTextField30.setBounds(new Rectangle(644, 146, 56, 19));
jTextField31.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField31.setText("1.00");
jTextField31.setBounds(new Rectangle(644, 167, 56, 19));
jTextField32.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField32.setText("10.01");
jTextField32.setBounds(new Rectangle(644, 189, 56, 19));
jTextField33.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField33.setText("-9.98");
jTextField33.setBounds(new Rectangle(644, 210, 56, 19));
jLabel27.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel27.setText("a0");
jLabel27.setBounds(new Rectangle(616, 150, 12, 15));
jLabel211.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel211.setText("b0");
jLabel211.setBounds(new Rectangle(616, 172, 12, 15));
jLabel212.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel212.setText("d0");
jLabel212.setBounds(new Rectangle(616, 216, 12, 15));
jLabel213.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel213.setText("c0");
jLabel213.setBounds(new Rectangle(616, 194, 12, 15));
```

```
jLabel214.setBounds(new Rectangle(619, 69, 28, 29));
jLabel214.setText("m");
jLabel214.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField37.setBounds(new Rectangle(644, 95, 56, 19));
jTextField37.setText("0.00");
jTextField37.setBorder(BorderFactory.createLineBorder(Color.black));
jLabel215.setBounds(new Rectangle(619, 91, 30, 29));
jLabel215.setText("R2");
jLabel215.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel216.setBounds(new Rectangle(619, 46, 30, 29));
jLabel216.setText("b1");
jLabel216.setFont(new java.awt.Font("Dialog", 1, 10));
jTextField36.setBounds(new Rectangle(644, 72, 56, 19));
jTextField36.setText("0.50");
jTextField36.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField35.setBounds(new Rectangle(644, 49, 56, 19));
jTextField35.setText("0.40");
jTextField35.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField34.setBounds(new Rectangle(644, 26, 56, 19));
jTextField34.setText("0.10");
jTextField34.setBorder(BorderFactory.createLineBorder(Color.black));
jLabel217.setBounds(new Rectangle(619, 22, 30, 29));
jLabel217.setText("a1");
jLabel217.setFont(new java.awt.Font("Dialog", 1, 10));
jPanel13.setBorder(border20);
jPanel13.setBounds(new Rectangle(608, 2, 110, 124));
jTextField41.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField41.setText("0.00");
jTextField41.setBounds(new Rectangle(755, 99, 46, 19));
jLabel219.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel219.setText("a2");
jLabel219.setBounds(new Rectangle(736, 28, 13, 15));
jTextField40.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField40.setText("0.20");
jTextField40.setBounds(new Rectangle(755, 73, 46, 19));
jTextField39.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField39.setText("0.83");
jTextField39.setBounds(new Rectangle(755, 49, 46, 19));
jTextField38.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField38.setText("-0.02");
jTextField38.setBounds(new Rectangle(755, 24, 46, 19));
jTextField29.setBorder(BorderFactory.createLineBorder(Color.black));
jTextField29.setText("0.00");
jTextField29.setBounds(new Rectangle(549, 364, 61, 19));
jLabel2112.setBounds(new Rectangle(396, 360, 143, 29));
jLabel2112.setText("Viscous Coupling Value");
jLabel2112.setFont(new java.awt.Font("Dialog", 1, 10));
saveInputData.setBounds(new Rectangle(662, 261, 124, 17));
saveInputData.setBorder(BorderFactory.createEtchedBorder());
saveInputData.setToolTipText("");
saveInputData.setActionCommand("Save Input Data");
saveInputData.setMnemonic('S');
saveInputData.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    saveInputData_actionPerformed(e);
  }
});
saveInputData.setText("Save Input Data");
openInputData.setText("Open Input Data");
openInputData.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(ActionEvent e) {
    openInputData_actionPerformed(e);
  }
});
openInputData.setMnemonic('O');
openInputData.setActionCommand("Save Input Data");
openInputData.setToolTipText("");
openInputData.setBorder(BorderFactory.createEtchedBorder());
openInputData.setBounds(new Rectangle(662, 281, 124, 17));
jProgressBar1.setBorder(border27);
jProgressBar1.setToolTipText("Simulation run status");
jProgressBar1.setMaximum(30);
jProgressBar1.setBounds(new Rectangle(641, 385, 173, 10));
```

```
jLabel220.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel220.setBorder(border28);
jLabel220.setText("        Simulation Run Status");
jLabel220.setBounds(new Rectangle(645, 375, 161, 13));
jLabel2110.setBounds(new Rectangle(736, 50, 15, 15));
jLabel2110.setText("b2");
jLabel2110.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel2111.setBounds(new Rectangle(736, 74, 13, 15));
jLabel2111.setText("n");
jLabel2111.setFont(new java.awt.Font("Dialog", 1, 10));
jLabel2113.setBounds(new Rectangle(736, 100, 13, 15));
jLabel2113.setText("R2");
jLabel2113.setFont(new java.awt.Font("Dialog", 1, 10));
jPanel14.setBounds(new Rectangle(720, 3, 97, 123));
jPanel14.setBorder(border18);
jPanel11.setBounds(new Rectangle(639, 240, 176, 131));
jPanel11.setBorder(border11);
implicitNewtonRaphson.setFont(new java.awt.Font("Dialog", 1, 10));
implicitNewtonRaphson.setBorder(null);
implicitNewtonRaphson.setSelected(false);
implicitNewtonRaphson.setMnemonic('0');
implicitNewtonRaphson.setText("N-Raphson ");
implicitNewtonRaphson.setVisible(true);
implicitNewtonRaphson.setBounds(new Rectangle(295, 255, 73, 13));
implicitNewtonJacobi.setFont(new java.awt.Font("Dialog", 1, 10));
implicitNewtonJacobi.setBorder(null);
implicitNewtonJacobi.setSelected(true);
implicitNewtonJacobi.setMnemonic('0');
implicitNewtonJacobi.setText("N-Jacobi");
implicitNewtonJacobi.setBounds(new Rectangle(295, 278, 73, 13));
implicitNewtonJacobi.setVisible(true);
jPanel15.setBorder(titledBorder29);
jPanel15.setVisible(true);
jPanel15.setBounds(new Rectangle(288, 234, 91, 68));
jPanel15.setBorder(titledBorder28);
jPanel16.setBorder(border37);
jPanel16.setBounds(new Rectangle(383, 342, 247, 56));
jif.getContentPane().add(jPanel1, null);
jPanel1.add(jLabel1, null);
jPanel1.add(jLabel2, null);
jPanel1.add(jLabel3, null);
jPanel1.add(jLabel4, null);
jPanel1.add(jLabel5, null);
jPanel1.add(jLabel6, null);
jPanel1.add(jLabel7, null);
jPanel1.add(jTextField1, null);
jPanel1.add(jTextField3, null);
jPanel1.add(jTextField2, null);
jPanel1.add(jTextField4, null);
jPanel1.add(jTextField5, null);
jPanel1.add(jTextField6, null);
jPanel1.add(jTextField7, null);
jPanel1.add(jTextField13, null);
jPanel1.add(jTextField12, null);
jPanel1.add(jTextField11, null);
jPanel1.add(jTextField10, null);
jPanel1.add(jTextField9, null);
jPanel1.add(jTextField8, null);
jPanel1.add(jLabel9, null);
jPanel1.add(jLabel10, null);
jPanel1.add(jLabel11, null);
jPanel1.add(jLabel12, null);
jPanel1.add(jLabel13, null);
jPanel1.add(jLabel14, null);
jPanel1.add(jButton5, null);
jif.getContentPane().add(jPanel3, null);
jPanel6.setBounds(new Rectangle(4, 18, 184, 62));
jPanel6.setLayout(null);
jPanel6.add(jTextField22, null);
jPanel6.setBorder(titledBorder6);
jPanel6.add(jLabel22, null);
jPanel6.add(r12, null);
jPanel6.add(jRadioButton2, null);
```

```
jPanel3.add(jPanel7, null);
jPanel7.add(jSlider1, null);
jPanel7.add(jTextField23, null);
jPanel7.add(jLabel23, null);
jPanel3.add(jPanel6, null);
jif.getContentPane().add(jPanel2, null);
jPanel2.add(jLabel17, null);
jPanel2.add(jLabel18, null);
jPanel2.add(jLabel19, null);
jPanel2.add(jLabel20, null);
jPanel2.add(jLabel21, null);
jPanel2.add(jLabel16, null);
jPanel2.add(jLabel15, null);
jPanel2.add(jLabel8, null);
jPanel2.add(jTextField16, null);
jPanel2.add(jTextField14, null);
jPanel2.add(jTextField15, null);
jPanel2.add(jTextField17, null);
jPanel2.add(jTextField18, null);
jPanel2.add(jTextField19, null);
jPanel2.add(jTextField20, null);
jPanel2.add(jTextField21, null);
jPanel9.add(no, null);
jPanel9.add(yes, null);
jif.getContentPane().add(saveInputData, null);
jif.getContentPane().add(openInputData, null);
jif.getContentPane().add(clearAll, null);
jif.getContentPane().add(stopSimulation, null);
jif.getContentPane().add(runSimulation, null);
jif.getContentPane().add(jPanel11, null);
jif.getContentPane().add(jLabel220, null);
jif.getContentPane().add(jProgressBar1, null);
jif.getContentPane().add(viscousCoupling, null);
jif.getContentPane().add(capillaryCoupling, null);
jif.getContentPane().add(bothCouplings, null);
jif.getContentPane().add(noCoupling, null);
jif.getContentPane().add(jPanel5, null);
jif.getContentPane().add(jTextField39, null);
jif.getContentPane().add(jTextField40, null);
jif.getContentPane().add(jLabel219, null);
jif.getContentPane().add(jTextField41, null);
jif.getContentPane().add(jTextField38, null);
jif.getContentPane().add(jLabel2110, null);
jif.getContentPane().add(jLabel2111, null);
jif.getContentPane().add(jLabel2113, null);
jif.getContentPane().add(jPanel14, null);
jif.getContentPane().add(jLabel210, null);
jif.getContentPane().add(jLabel29, null);
jif.getContentPane().add(jLabel28, null);
jif.getContentPane().add(jTextField26, null);
jif.getContentPane().add(jTextField27, null);
jif.getContentPane().add(jTextField28, null);
jif.getContentPane().add(jPanel10, null);
jif.getContentPane().add(jTextField30, null);
jif.getContentPane().add(jLabel213, null);
jif.getContentPane().add(jTextField31, null);
jif.getContentPane().add(jTextField32, null);
jif.getContentPane().add(jTextField33, null);
jif.getContentPane().add(jLabel212, null);
jif.getContentPane().add(jLabel211, null);
jif.getContentPane().add(jLabel27, null);
jif.getContentPane().add(jPanel8, null);
jif.getContentPane().add(jTextField35, null);
jif.getContentPane().add(jLabel217, null);
jif.getContentPane().add(jTextField34, null);
jif.getContentPane().add(jLabel216, null);
jif.getContentPane().add(jLabel214, null);
jif.getContentPane().add(jTextField36, null);
jif.getContentPane().add(jTextField37, null);
jif.getContentPane().add(jLabel215, null);
jif.getContentPane().add(jPanel13, null);
jif.getContentPane().add(jPanel4, null);
jPanel4.add(jLabel24, null);
```

```
 jPanel4.add(jLabel25, null);
 jPanel4.add(jTextField24, null);
 jPanel4.add(jTextField25, null);
 jif.getContentPane().add(jPanel9, null);
 R12.add(r12);
 R12.add(jRadioButton2);
 Flow.add(no);
 Flow.add(yes);
 Coupling.add(viscousCoupling);
 Coupling.add(capillaryCoupling);
 Coupling.add(bothCouplings);
 Coupling.add(noCoupling);
 jif.getContentPane().add(implicitNewtonJacobi, null);
 jif.getContentPane().add(implicitNewtonRaphson, null);
 jif.getContentPane().add(jPanel15, null);
 jif.getContentPane().add(jTextField29, null);
 jif.getContentPane().add(jLabel2112, null);
 jif.getContentPane().add(jPanel16, null);
 SolutionMethod.add(implicitNewtonRaphson);
 SolutionMethod.add(implicitNewtonJacobi);
 jif.getContentPane().add(jScrollPane1);
 jScrollPane1.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
 jScrollPane1.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
 jif.setClosable(true);
 jif.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
 jif.setResizable(false);
 jif.setOpaque(true);
}
//Initiation of GUI ends here

// A method to set all textfields in the fluid properties box to null, i.e. " "
void jButton5_actionPerformed(ActionEvent e) {
 for(int i = 1; i < (13+1); i++){
  jTextFieldArray[i].setText("");
 }
}

//A method to stop simulation run.
void stopSimulation_actionPerformed(ActionEvent e) {

 //Stop simulation run by setting isStop to true using a annonymous thread class
 if(isStop == false){
  new Thread(new Runnable() {
   public void run() {
    isStop = true;
   }
  }).start();

  //Reset all relevant static variables
  flowCount = 0;
  residualValueSum = 0;
  timeDivision = 0;
  flowIterationNumber = 0;
  jif.setTitle(" ");
  for(int i = 0; i < bNum; i++){
   copyE[i] = 0;
  }

  //Reset the progressbar to signify end of computation
  jProgressBar1.setIndeterminate(false);

  //Message box to signify end of simulation, when the simulation is interrupted
  JOptionPane.showMessageDialog(null,"Simulation has been interrupted. Try again!");
 }
}

//A method to set all textfields to null, i.e. " "
void clearAll_actionPerformed(ActionEvent e) {
 for(int i = 1; i < (41+1); i++){
  jTextFieldArray[i].setText("");
 }
}
```

```
//A method to obtain the current value of jSlider1 and put it in valueOfSlide
//and then display the current value of valueOfSlide in the jTextField23
 void jSlider1_actionChange(ChangeEvent e){
   valueOfSlide = jSlider1.getValue();
   jTextField23.setText(String.valueOf(valueOfSlide));
  }

//A method to obtain the current value of jTextField23 and put it in valueOfSlide
//and then display the current value of valueOfSlide in the jSlider1.
//This is the reverse of the above method.
void  jTextField23_actionPerformed(ActionEvent e){
  valueOfSlide = Integer.parseInt(jTextField23.getText());
  jSlider1.setValue(valueOfSlide);
}

//Method to set value of a for R12 computation = 0.
//This option leads to R12 = 1, since R12 = (1 - a * (1-S))
//Also set the TextField holding a uneditable. In addition specify
//char = '0', to be saved as input data
void jRadioButton2_actionPerformed(ActionEvent e) {
  jTextField22.setText("0.00");
  jTextField22.disable();
  r12Selection = '0';
}


// Method to set value of "a" for R12 computation != 0.
//That is R12 should be computed. This option also resets "a" = 0.05.
// Also set the TextField holding "a" editable. In addition specify
//char = '1', to be saved as input data
void r12_actionPerformed(ActionEvent e) {
  jTextField22.setText("0.05");
  jTextField22.enable();
  r12Selection = '1';
}


 // Method (Option) to enable real-time toggling of solution options (N-Jacobi or N-Raphson)
 void no_actionPerformed(ActionEvent e) {
   implicitNewtonRaphson.setEnabled(false);
   implicitNewtonJacobi.setEnabled(false);
 }

  // Method (Option) to disenable real-time toggling of solution options (N-Jacobi or N-Raphson)
  void yes_actionPerformed(ActionEvent e) {
   implicitNewtonRaphson.setEnabled(true);
   implicitNewtonJacobi.setEnabled(true);
  }

//This method makes the TextFields holding the values of VisCou uneditable
// It also sent the content to 1
void noCoupling_actionPerformed(ActionEvent e) {
  jTextField29.setText("0.0");
  jTextField29.disable();
  couplingSelection = 'n';
}

//The next 3 methods make the TextFields holding the values of VisCou editable
void bothCouplings_actionPerformed(ActionEvent e) {
  jTextField29.enable();
  couplingSelection = 'b';
  jTextField29.setText("2.00");
}

void capillaryCoupling_actionPerformed(ActionEvent e) {
  jTextField29.disable();
  couplingSelection = 'c';
  jTextField29.setText("0.0");
}

void viscousCoupling_actionPerformed(ActionEvent e) {
  jTextField29.enable();
  couplingSelection = 'v';
  jTextField29.setText("2.00");
}

 void relative_actionPerformed(ActionEvent e) {
  noCoupling.setSelected(true);
  jTextField29.setText("0.0");
  jTextField29.disable();
  couplingSelection = 'n';
 }


//CONSTRUCTION OF GUI ENDS HERE

//DECLARATION OF ALL PUBLIC VARIABLES STARTS HERE

   /** Variables (fields) to hold input from TextFields
   * nwDensity = non-wetting Density, wDensity = wetting Density, nwVis = non-wetting viscosity
   * wVis = wetting viscosity, injTime = injection time, wInjRate = wetting injection time
   * fluidIterationNumber =  number of time interval to compute fluid properties
   * effKnw = effective permeability to non-wetting at Swi (This is reference permeability for Kr)
   * effKw = effective permeability to wetting at Sor
   * Swi = initial or irreducible wetting saturation
   * Sor = residual or irreducible non-wetting saturation
   * wInjRate  = wetting injection Rate
   * flowTolerance = tolerance flow fractional flow solution convergence
   * flowCountTolerance = maximum number of iteration before exception is generated
   * w = wetting e.g. water, nw = non-wetting e.g. oil
   * l = Length, t = thickness, h = height, pv = pore volume, inc = inclination
   * Pd = displacement pressure, Ac = Area under the capillary pressure curve
   * R12a = hydrodynamic effect factor, bNum = number of saturation grid/block
   * Swf = breakthrough saturation, Fwf = breakthrough fractional flow
   * VisCou = ViscousCoupling term
   * pca0, pca1, pca2, pca3 = Pc fitting coeeficients
   * krwa0, krwa1, n, krwa_R2, kroa0, kroa1, m, kroa_R2 = Kr fitting cofficients
   * InitialF = Initially assumed solution of Frac. Flow in each grid block
   * InletF = Inlet Frac. Flow, InletS_a = coeff. a for S* computation = 1.0
   * InletS_b = coeff. b for S* computation, InletS_c = coeff. c for S* computation
   * InletS_d = coeff. d for S* computation, InletS = Initially Guessed Inlet Sat.
   * Note: w = wetting (water) and o = non-wetting (oil)
   */

   double nwDensity, wDensity, nwVis, wVis, injTime, wInjRate, flowTolerance;
   int flowIterationNumber, bNum, flowCountTolerance;
   double effKnw, effKw, Swi, Sor;
   double l, t, h, pv, porosity, inc, Pd, Ac;
   double R_12a, Swf, Fwf, InletSAdjust, InletSGuess, VisCou;
   double pca0, pca1, pca2, pca3;
   double krwa0, krwa1, m, krwa_R2, kroa0, kroa1, n, kroa_R2;
   double InitialF, InletF, InitialDis;
   double InletS_a = 1.0; double InletS_b, InletS_c, InletS_d;

   //Variable to save coupling selection option. The default data is viscous 'v'
   char couplingSelection  = 'v';

   //Variable to select r12 option. The default data is do not select R12 '0'
   char r12Selection = '0';

   // A public variable to be used by jSlider1_actionChange and jTextField23_actionPerformed
   int valueOfSlide;

  //A static variable to count the number of new "input data"
   static int inputNumber = 0;

   //A static variable to count the number of Flow Solution Iteration"
   static int flowCount = 0;

   //Array to hold computed distance travelled
   double [] normalizedDis;

   //Arrays to hold computed pressure gradient
   double [] norPressPotenOne, norPressPotenTwo;

   //Variable for residual value computation
```

```
    double residualValueSum;

    //Variable for monitoring simulation time interval
    double initialTime, finalTime;

    //Variables to hold MBE values
    double mbeResult;
    static double sumOfMBE = 0;

    //Variable for time interval division
    static int timeDivision;

    //Variable to count the number of new "output results"
    static int ouputNumber = 0;

    //Copies of the solution Matrix for iterative purpose
    double [] copyE = new double [bNum];
    double [] copyETime = new double [bNum];
    double [] copyETimeTime = new double [bNum];
    double [] copyDisTime = new double [bNum];
    double [] Dii = new double [bNum];
    double [] fOfSS = new double [bNum];
    double [] gOfSS = new double [bNum];
    double [] cOfSS = new double [bNum];
    double [] xettaOfSS = new double [bNum];
    double [] ss = new double [bNum];

   //Create JTextField Array for opening save input data
    JTextField [] jTextFieldArray = {jTextField1, jTextField1, jTextField2, jTextField3,
      jTextField4, jTextField5, jTextField6, jTextField7, jTextField8, jTextField9, jTextField10,
      jTextField11, jTextField12, jTextField13, jTextField14, jTextField15, jTextField16, jTextField17,
      jTextField18, jTextField19, jTextField20, jTextField21, jTextField22, jTextField23, jTextField24,
      jTextField25, jTextField26, jTextField27, jTextField28, jTextField29, jTextField30, jTextField31,
      jTextField32, jTextField33, jTextField34, jTextField35, jTextField36, jTextField37, jTextField38,
      jTextField39, jTextField40, jTextField41};

//DECLARATION OF ALL PUBLIC VARIABLES STOPS HERE

//IMPLEMENTATION OF "Normalized and other variables" STARTS HERE

  public double mr(){
    double mr = ((effKw * nwVis)/(effKnw * wVis));
    return mr;
   }

   public double nc(){
    double nc = (Ac * (effKw/wVis))/(velocity()*l);
    return nc;
   }

    public double ng(){
    double ng = (((effKw/wVis) * deltaDensityRho() * 9.81) * Math.sin((Math.PI * inc/180)))/velocity();
    return ng;
   }

   public double deltaDensityRho(){
    double deltaDensityRho = (wDensity - nwDensity);
    return  deltaDensityRho;
   }

   public double rOneTwo(double s){
    double rOneTwo = (1 - (R_12a*(1-s)));
    return rOneTwo;
   }

   public double lamdaOne(double s){
    double lamdaOne = (krwValue(s) * effKw / wVis);
    return lamdaOne;
   }

   public double lamdaTwo(double s){
    double lamdaTwo = (kroValue(s) * effKnw /nwVis);
    return lamdaTwo;
  }

  public double lamdaM(double s){
   double lamdaM = (s * (effKw/wVis)) + ((1 - s)* (effKnw/nwVis)) ;
   return lamdaM;
  }

  public double [] alphaOne(){
    double [] A = new double [bNum];

    for(int i = 0; i < bNum; i++){
     //No coupling
     if(noCoupling.isSelected()){
       A[i] = 1 * 1;
     }
     //Only capillary coupling
     else if(capillaryCoupling.isSelected()){
       A[i] = (1 - porosity);
     }
     //Only Viscous coupling
     else if(viscousCoupling.isSelected()){
       A[i] = ((1 - ((VisCou / rOneTwo((i+1) * deltaS())) * (lamdaTwo((i+1) * deltaS())/lamdaM((i+1)*deltaS())))));
     }
     //Both viscous and capillary coupling
     else if(bothCouplings.isSelected()){
       A[i] = (((1 - porosity) * ((1 - (VisCou / rOneTwo((i+1) * deltaS())) * (lamdaTwo((i+1) * deltaS())/lamdaM((i+1)*deltaS()))))));
     }
    }
    return A;
  }

  public double [] alphaTwo(){
    double [] A = new double [bNum];

    for(int i = 0; i < bNum; i++){
     //No coupling
     if(noCoupling.isSelected()){
       A[i] = 1 * 1;
     }
     //Only capillary coupling
     else if(capillaryCoupling.isSelected()){
       A[i] = (1 - porosity);
     }
     //Only Viscous coupling
     else if(viscousCoupling.isSelected()){
       A[i] = ((1 - (VisCou * rOneTwo((i+1) * deltaS())) * (lamdaOne((i+1) * deltaS())/lamdaM((i+1) * deltaS()))));
     }
     //Both viscous and capillary coupling
     else if(bothCouplings.isSelected()){
       A[i] = ((1 - porosity) * ((1 - (VisCou * rOneTwo((i+1) * deltaS())) * (lamdaOne((i+1) * deltaS())/lamdaM((i+1) * deltaS())))));
     }
    }
    return A;
  }

  public  double [] fOfS() {
   double [] A = new double [bNum];
   for (int i = 0; i < bNum; i++){
    if(noCoupling.isSelected() && R_12a == 0)
     A[i] = (mr() * krwValue((i+1)*deltaS())) / ( mr() * krwValue((i+1)*deltaS()) + kroValue((i+1)*deltaS()));
    else if(noCoupling.isSelected())
     A[i] = (rOneTwo((i+1)*deltaS()) * mr() * krwValue((i+1)*deltaS())) / (rOneTwo((i+1)*deltaS()) * mr() * krwValue((i+1)*deltaS())
+ kroValue((i+1)*deltaS()));
    else
     A[i] = (rOneTwo((i+1)*deltaS()) * mr() * krwValue((i+1)*deltaS())) / (rOneTwo((i+1)*deltaS()) * mr() * krwValue((i+1)*deltaS())
+ kroValue((i+1)*deltaS()));
   }
  return A;
  }

  public double [] gOfS(){
   double [] A = new double [bNum];
   for (int i = 0; i < bNum; i++){
      if(noCoupling.isSelected() && R_12a == 0)
        A[i] = (1 - (ng() * kroValue((i+1)*deltaS()) / mr())) * fOfS()[i];
      else if(noCoupling.isSelected())
        A[i] = (1 - (1 *((1 + ((nwDensity/deltaDensityRho()) * (1 - rOneTwo((i+1)*deltaS())))) * ng() *
kroValue((i+1)*deltaS()))/(rOneTwo((i+1)*deltaS()) * mr())))*fOfS()[i];
      else
        A[i] = (1 - (couplingTerm()[i] *((1 + ((nwDensity/deltaDensityRho()) * (1 - rOneTwo((i+1)*deltaS())))) * ng() *
kroValue((i+1)*deltaS()))/(rOneTwo((i+1)*deltaS()) * mr())))*fOfS()[i];
    }
    return A;
  }

  public double [] cOfS(){
    double [] A = new double [bNum];
    for (int i = 0; i < bNum; i++){
      if(noCoupling.isSelected() && R_12a == 0)
        A[i] = -((1/mr()) * fOfS()[i] * kroValue((i+1)*deltaS()) * deltaPlc((i+1)*deltaS()));
      else if(noCoupling.isSelected())
        A[i] = (-1/(mr() * rOneTwo((i+1) * deltaS())))*(1 * fOfS()[i] * kroValue((i+1)*deltaS()) * deltaPlc((i+1)*deltaS()));
      else
        A[i] = (-1/(mr() * rOneTwo((i+1) * deltaS())))*(couplingTerm()[i] * fOfS()[i] * kroValue((i+1)*deltaS()) *
deltaPlc((i+1)*deltaS()));
     }
    return A;
  }

  public double norTime(){
    double norTime = (velocity() * injTime) / (porosity * l * (1 - Sor - Swi));
    return norTime;
  }

  public double norFullTime(){
    double norFullTime = (norTime()/flowIterationNumber);
    return norFullTime;
  }

  public double nettaOfS(){
    double A = ((norFullTime())/(nc() * Math.pow(deltaS(),2)));
    return A;
  }

  public double [] xettaOfS(){
    double [] A = new double [bNum];
    for (int i = 0; i < bNum; i++){
       A[i] = (nettaOfS()/cOfS()[i]);
    }
     return A;
  }

  public double velocity(){
    double velocity = (wInjRate)/(t * h);
    return velocity;
  }

  public double inletSActual(){
    double inletS = 0;
    double a = Math.pow((InletS_c * norTime()/ nc()), InletS_d);
    double b = InletS_b * Math.exp(-a);

    if(InletS_b == 0 && InletS_c == 0){
      inletS = InletS_d;
    }
    else{
      inletS = InletS_a * (1 - b);
    }
    return inletS;
 }

  public double deltaS(){
     return (0.99999999999/bNum) * Math.abs(inletSActual());
 }

//IMPLEMENTATION OF "Normalized and other Variables" ENDS HERE
```

```
//IMPLEMENTATION OF "Kr", "Coupling Term" and "dZetta/dS" STARTS HERE

   public double deltaPlc(double s){
    double deltaPlc = deltaPlc = (1/Ac) * ( ( (((1 + pca2*s + pca3*s*s)*(-pca0-2*pca1*s))) - (((pca0*(1-s) + pca1*(1-s*s))*(pca2 +
2*pca3*s))) ) /  ((1 + pca2*s + pca3*s*s)*(1 + pca2*s + pca3*s*s)) );
    return deltaPlc;
   }

   public double krwValue(double s){
    double krwValue = ((krwa0 + krwa1 * (1-s)) / (krwa0 + (1-s))) * Math.pow(s, m);
    return krwValue;
   }

  public double kroValue(double s){
   double kroValue = ((kroa0 + kroa1 * (s)) / (kroa0 + (s))) * Math.pow((1-s), n);
   return kroValue;
   }

  public double [] couplingTerm(){
   double [] A = new double [bNum];
   for (int i = 0; i < bNum; i++){
    A[i] = ((alphaOne()[i] + alphaTwo()[i])/2);
   }
   return A;
  }

 //IMPLEMENTATION OF "Kr", "Coupling Term" and "dZetta/dS"  ENDS HERE

 //IMPLEMENTATION OF "Flow Coefficients, distance and pressure computations" STARTS HERE

 public double [] Di(double [] gOfS, double [] xettaOfS){
    double [] A = new double[bNum];

    //Populate A
    for(int i = 0; i < bNum; i++){
      if(flowCount == 1)
          A[i] = (Math.pow((InletF - gOfS[i]),2) * (InletF - 2*InletF + InletF) * xettaOfS[i] - InletF + IC()[i]);
      else if (flowCount > 1)
        if(i == 0)
          A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * (copyE[i+1] - 2*copyE[i] + 0) * xettaOfS[i] - copyE[i] + IC()[i]);
        else if(i > 0 && i < (bNum-1))
          A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * (copyE[i+1] - 2*copyE[i] + copyE[i-1]) * xettaOfS[i] - copyE[i] + IC()[i]);
        else if(i == (bNum-1))
          A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * (0 - 2*copyE[i] + copyE[i-1]) * xettaOfS[i] - copyE[i] + IC()[i]);
    }
    return A;
  }

  public double [] dFplusOne(double [] gOfS, double [] xettaOfS){
   int j = bNum-1;
   double [] A = new double[j];

   //Populate A
   for(int i = 0; i < j; i++){
    if(flowCount == 1){
     A[i] = (Math.pow( (InletF - gOfS[i]), 2) * xettaOfS[i]);
    }
    else if(flowCount > 1){
     if(i < j-1)
      A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * xettaOfS[i]);
     else
      A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * xettaOfS[i]);
    }
   }
   return A;
  }

  public double [] dFminusOne(double [] gOfS, double [] xettaOfS){
   int j = bNum-1;
   double [] A = new double[j];

   //Populate A
   for(int i = 0; i < j; i++){
    if(flowCount == 1){
     A[i] = (Math.pow( (InletF - gOfS[i]), 2) * xettaOfS[i]);
    }
    else if(flowCount > 1){
     if(i < j-1)
      A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * xettaOfS[i]);
     else
      A[i] = (Math.pow((copyE[i] - gOfS[i]),2) * xettaOfS[i]);
    }
   }
   return A;
  }

   public double [] dFnone(double [] gOfS, double [] xettaOfS){
    int j = bNum;
    double [] A = new double[j];

    //Populate A
    for(int i = 0; i < j; i++){
     if(flowCount == 1)
        A[i] = (2 * xettaOfS[i]*(InletF - gOfS[i])*(InletF - 3*InletF + InletF + gOfS[i]))-1;
     else if (flowCount > 1)
        if(i == 0)
         A[i] = (2 * xettaOfS[i]*(copyE[i] - gOfS[i])*(copyE[i+1] - 3*copyE[i] + 0 + gOfS[i]))-1;
        else if(i > 0 && i < (j-1))
         A[i] = (2 * xettaOfS[i]*(copyE[i] - gOfS[i])*(copyE[i+1] - 3*copyE[i] + copyE[i-1] + gOfS[i]))-1;
        else if(i == (j-1))
         A[i] = (2 * xettaOfS[i]*(copyE[i] - gOfS[i])*(0 - 3*copyE[i] + copyE[i-1] + gOfS[i]))-1;
    }
    return A;
   }

  //Normalized distance travelled
  public double [] zetta(double f [], double ff[], double PrevDis [], double s, int timeDivisionn){
    int j = f.length;

    double [] A = new double [j];
    double c = norFullTime()/s;  //ratio of deltaTime (norFullTime) to deltaS

    //Compute individual grid distance based on flow
    for(int i = 0; i < j-1; i++) {
     if(timeDivisionn == 1)
      if(i == 0)
       A[i] = ( (f[i] - 0) + (ff[i] - 0) ) * c/2 + InitialDis;
      else if(i > 0 && i < j-1)
       A[i] = ( (f[i] - f[i-1]) + (ff[i] - ff[i-1]) ) * c/2 + InitialDis;
      else
       A[i] = ( (f[i] - f[i-1]) + (ff[i] - ff[i-1]) ) * c/2 + InitialDis;
     else if(timeDivisionn > 1)
      if(i == 0)
       A[i] = ( (f[i] - 0) + (ff[i] - 0) ) * c/2 + PrevDis[i];
      else if(i > 0 && i < j-1)
       A[i] = ( (f[i] - f[i-1]) + (ff[i] - ff[i-1]) ) * c/2 + PrevDis[i];
      else
       A[i] = ( (f[i] - f[i-1]) + (ff[i] - ff[i-1]) ) * c/2 + PrevDis[i];
    }

   return A;
  }

  //Normalized potential gradient in the wetting phase
  public double [] potentialGradientOne(double [] fSol, double [] dis, double s){
   // fSol = fractional flow
   // dis = Normalized distance travelled
   // s = deltaS
   double[] C = new double[bNum];   // Saturation gradient
   double[] E = new double[bNum];   // Potential gradient in the non-wetting phase
   double A3 = 0;
   double A4 = 0;

   //Compute saturation gradient
   for(int i = 0; i < bNum; i++) {
    if(i == 0)
      C[i] =  s/(dis[i] - (- dis[1] + 2*dis[0] ) );  //based on extrapolated distance
    else if(i > 0 && i < bNum-1)
      C[i] =  0.5*s/(dis[i+1] - dis[i-1]);           //centre diff.
    else if(i == (bNum-1))
      C[i] =  s/(dis[i] - dis[i-1]);                 //backward diff. to handle boundary value
   }

   for(int i = 0; i < bNum; i++){
    if(noCoupling.isSelected() && R_12a == 0){
      E[i] =  - (velocity() * fSol[i]) / lamdaOne( (i+1)*s );
    }
     else if(noCoupling.isSelected()){
      E[i] =  - (velocity() * fSol[i]) / lamdaOne( (i+1)*s ) ;
    }
     else{
      A3 =  -(velocity() * fSol[i]) / lamdaOne( (i+1)*s ) ;
      A4 = ( (1 - alphaOne()[i]) / (2 * 1) ) *
         ( ( deltaPlc((i+1)*s) * Ac * -C[i] ) - ((wDensity - rOneTwo((i+1)*s)* nwDensity) * 9.81 * Math.sin((Math.PI * inc/180)) ) );
      E[i] = (A3 - A4);
     }
    }
    return E;
  }

  //Normalized potential gradient in the non-wetting phase
  public double [] potentialGradientTwo(double [] fSol, double [] dis, double s){
   // dis = Normalized distance travelled
    // s = deltaS
    double[] C = new double[bNum]; // Saturation gradient
    double[] E = new double[bNum]; // Potential gradient in the wetting phase
    double A3 = 0;
    double A4 = 0;

    //Compute saturation gradient
    for(int i = 0; i < bNum; i++) {
      if(i == 0)
        C[i] =  s/(dis[i] - (- dis[1] + 2*dis[0] ) );  //based on extrapolated distance
      else if(i > 0 && i < bNum-1)
        C[i] =  0.5*s/(dis[i+1] - dis[i-1]);           //centre diff.
      else if(i == (bNum-1))
        C[i] =  s/(dis[i] - dis[i-1]);                 //backward diff. to handle boundary value
    }

   for(int i = 0; i < bNum; i++){
    if(noCoupling.isSelected() && R_12a == 0){
      E[i] =  -(velocity() * (1 - fSol[i])) / lamdaTwo( (i+1)*s);
    }
    else if(noCoupling.isSelected()){
      E[i] =  - ( (velocity() * (1 - fSol[i])) / lamdaTwo( (i+1)*s) );
   }
    else{
      A3 =  -(velocity() * (1 - fSol[i])) / lamdaTwo( (i+1)*s ) ;
      A4 = ( (1 - alphaTwo()[i]) / (2 * rOneTwo((i+1)*s)) ) *
         ( ( deltaPlc((i+1)*s) * Ac * -C[i] ) - ((wDensity - rOneTwo((i+1)*s)* nwDensity) * 9.81 * Math.sin((Math.PI * inc/180)) ) );
      E[i] = (A3 + A4);
     }
   }
   return E;
  }

  //Discretized saturation for dynamic viewing
  public double [] sat(){
    double [] sat = new double [bNum];
    for(int i = 0; i < bNum; i++){
      sat[i] = (i+1) * deltaS();
    }
    return sat;
  }

//IMPLEMENTATION OF "Flow Coefficients, distance and pressure computations" ENDS HERE

//A METHOD FOR COPYING ARRAY STARTS HERE
```

```
 public double [] copy(double [] A){
  int j = A.length;
  double [] E = new double [j];
  for(int i = 0; i < j; i++){
   E[i] = A[i];
  }
  return E;
 }

//A METHOD FOR COPYING ARRAY STOPS HERE

//A METHOD USED IN COMPUTING NAIVE-JACOBI-NEWTON SOLUTION STARTS HERE

 public double [] newtonJacobi(double [] A, double [] B){
  int j = A.length;
  double [] E = new double[j];
  for(int i = 0; i < j; i++){
    E[i] = A[i]/B[i];
   }
  return E;
 }

//A METHOD USED IN COMPUTING NAIVE-JACOBI-NEWTON SOLUTION STOPS HERE

//A METHOD FOR FLOW ACTUAL SOLUTION STARTS HERE

 public double [] flowActualSolution(double [] A,  int flowCount){
  int j = A.length;
  double [] E = new double [j];
  double initialAssumption = InletF;

  if(flowCount == 1){
   for(int i = 0; i < j; i++){
    E[i] = initialAssumption - A[i];
   }
  }
  else if(flowCount > 1){
   for(int i = 0; i < j; i++){
    if(i < (j-1)){
     E[i] = copyE[i] - A[i];
    }
    else if(i == (j-1)){
     if(implicitNewtonJacobi.isSelected()){
      E[i] = 1;
     }
     else if(implicitNewtonRaphson.isSelected()){
      E[i] = copyE[i] - A[i];
     }
    }
   }
  }
  return E;
 }

//A METHOD FOR FLOW ACTUAL SOLUTION ENDS HERE

//A METHOD FOR INDIVIDUAL RESIDUAL FLOW COMPUTATION STARTS HERE

 public double residual(double [] A, int flowCount){
  int j = A.length;
  double [] R = new double [j];

  if(flowCount == 1){
   for(int i = 0; i < j; i++){
    R[i] = Math.abs(A[i] - InletF);
   }
  }
  else if(flowCount > 1){
   for(int i = 0; i < j; i++){
    R[i] = Math.abs(A[i] - copyE[i]);
   }
  }
```

```
   return maximumOfArrayValues(R);
  }

//A METHOD FOR INDIVIDUAL RESIDUAL FLOW COMPUTATION ENDS HERE

//A METHOD FOR MBE COMPUTATION STARTS HERE

 public double calculateMBE(double [] distance){
   int j = distance.length + 1;
   double distancee [] = new double [j];
    sumOfMBE = 0;

   for(int i = 0; i < j; i++){
     if(i == 0){
       distancee[i] = - distance[1] + 2*distance[0]; //extrapolate last distance
     }
     else{
       distancee[i] = distance[i-1];
     }
   }

   for(int i = 0; i < j; i++){
     if(i < (j-1)){
       sumOfMBE = sumOfMBE + deltaS()*0.5*(distancee[i] + distancee[i+1]);
     }
     else if(i == (j-1)){
       sumOfMBE = sumOfMBE + deltaS()*0.5*(distancee[i] + 0);
     }
   }

   double calculateMBE = Math.abs(norTime() - sumOfMBE);

   return calculateMBE;
  }

//A METHOD FOR MBE COMPUTATION ENDS HERE


//A METHOD FOR RETURNING MAXIMUM VALUE FROM A 1-D ARRAY OF DOUBLE VALUE STARTS HERE

  static double maximumOfArrayValues(double [] A){
   double staticMaximumValue = 0;
   double dynamicMaximumValue = 0;
   for(int i = 0; i < (A.length - 1); i++){
     if(i == 0){
       if(A[i] >= A[i+1]){
         dynamicMaximumValue = A[i];
         staticMaximumValue = dynamicMaximumValue;
       }
       else{
         dynamicMaximumValue = A[i+1];
         staticMaximumValue = dynamicMaximumValue;
       }
     }
     else if(i > 0){
       if(A[i] >= A[i+1]){
         dynamicMaximumValue = A[i];
         staticMaximumValue = Math.max(staticMaximumValue, dynamicMaximumValue);
       }
       else{
         dynamicMaximumValue = A[i+1];
         staticMaximumValue = Math.max(staticMaximumValue, dynamicMaximumValue);
       }
     }
   }
   double LastMaximumValue = staticMaximumValue;
   return LastMaximumValue;
  }

//A METHOD FOR RETURNING MAXIMUM VALUE FROM A 1-D ARRAY OF DOUBLE VALUE ENDS HERE

//A METHOD USED IN INITIALIZING InletF (BC) IN ALL FLOW COMPUTATIONS STARTS HERE
```

```
  public double [] IC(){
    double [] E = new double [bNum];
    double flowBoundaryCondition = InitialF;

    if(timeDivision == 1)
      for(int i = 0; i < bNum; i++){
        E[i] = flowBoundaryCondition;
      }
    else if(timeDivision > 1)
      for(int i = 0; i < bNum; i++){
        if(i < (bNum-1))
         E[i] = copyETime[i];
        else
         E[i] = flowBoundaryCondition;
      }
    return E;
  }
//A METHOD USED IN INITIALIZING InletF (BC) IN ALL FLOW COMPUTATIONS ENDS HERE

//ACTUAL COMPUTATION AND SIMULATOR IMPLEMENTATION STARTS HERE

  //A Method to run all the codes/classes made available here
  void runSimulation_actionPerformed(ActionEvent e) {

    //Get and store the current title of jif for dynamic update of simulation status
    updateTitle = jif.getTitle();
    updateTitleNow = updateTitle;

    //Re-zero relevant variables
    flowCount = 0;
    for(int i = 0; i < bNum; i++){
     copyE[i] = 0;
    }

    //Statement to ignore if code is still run
    if(isStop == true){

     //Set isStop false to allow thread to run. This is necessary incase
     //simulation is stopped before it completes running
     isStop = false;

     //Annonymous Thread class to handle computation on a thread
     new Thread(new Runnable() {

       //Implementation of run method of the runnable interface
       public void run() {

         //Set progress bar to indicate computation in progess
         jProgressBar1.setIndeterminate(true);

         //Time at the begining of simulation in seconds
         initialTime = (System.currentTimeMillis() / 1000);

         try {
           /**Passing TextFields' values to actual variables (fields)*/

           nwDensity = Double.parseDouble(jTextField1.getText());
           wDensity = Double.parseDouble(jTextField2.getText());
           nwVis = Double.parseDouble(jTextField3.getText());
           wVis = Double.parseDouble(jTextField4.getText());
           injTime = Double.parseDouble(jTextField5.getText());
           wInjRate = Double.parseDouble(jTextField6.getText());
           flowIterationNumber = Integer.parseInt(jTextField7.getText());
           effKnw = Double.parseDouble(jTextField8.getText());
           effKw = Double.parseDouble(jTextField9.getText());
           Swi = Double.parseDouble(jTextField10.getText());
           Sor = Double.parseDouble(jTextField11.getText());
           flowCountTolerance = Integer.parseInt(jTextField12.getText());
           flowTolerance = Double.parseDouble(jTextField13.getText());
           l = Double.parseDouble(jTextField14.getText());
           t = Double.parseDouble(jTextField15.getText());
           h = Double.parseDouble(jTextField16.getText());
           pv = Double.parseDouble(jTextField17.getText());
           porosity = Double.parseDouble(jTextField18.getText());
           inc = Double.parseDouble(jTextField19.getText());
           Pd = Double.parseDouble(jTextField20.getText());
           Ac = Double.parseDouble(jTextField21.getText());
           R_12a = Double.parseDouble(jTextField22.getText());
           bNum = Integer.parseInt(jTextField23.getText());
           InitialDis = Double.parseDouble(jTextField24.getText());
           InletF = Double.parseDouble(jTextField25.getText());
           InitialF = Double.parseDouble(jTextField25.getText());
           InletS_b = Double.parseDouble(jTextField26.getText());
           InletS_c = Double.parseDouble(jTextField27.getText());
           InletS_d = Double.parseDouble(jTextField28.getText());
           VisCou = Double.parseDouble(jTextField29.getText()) * porosity * porosity;
           pca0 = Double.parseDouble(jTextField30.getText());
           pca1 = Double.parseDouble(jTextField31.getText());
           pca2 = Double.parseDouble(jTextField32.getText());
           pca3 = Double.parseDouble(jTextField33.getText());
           krwa0 = Double.parseDouble(jTextField34.getText());
           krwa1 = Double.parseDouble(jTextField35.getText());
           m = Double.parseDouble(jTextField36.getText());
           krwa_R2 = Double.parseDouble(jTextField37.getText());
           kroa0 = Double.parseDouble(jTextField38.getText());
           kroa1 = Double.parseDouble(jTextField39.getText());
           n = Double.parseDouble(jTextField40.getText());
           kroa_R2 = Double.parseDouble(jTextField41.getText());

           //Compute only relevant coefficients once
           jif.setTitle(updateTitleNow + ".  Computing coefficients.....");

           //Compute relevant coefficients
              fOfSS = fOfS();
              gOfSS = gOfS();
              cOfSS = cOfS();
              xettaOfSS = xettaOfS();
              ss = sat();

           //Set initial state of "FlowView" object
           f1.getContentPane().add(fv1);
           f2.getContentPane().add(fv2);
         }

         catch (NumberFormatException nfe) {
           //Exception to catch non-numeric or missing values in the TextFields
           JOptionPane.showMessageDialog(null, "Missing or Non-Numeric Input Value(s).");
         }

         try {
           /**Solve flow and displacement equations with do-while
            * using Newton-Raphson or Newton-Jacobi Solution method
            * condition respectively
            */

            do {
              timeDivision = timeDivision + 1;

              do {
                //Solve the flow equation using ThomasAlgorithmSolution.
                //Compute residual and iterate untill condition is met.

                flowCount = flowCount + 1; // This static variable is related to residualValueSum

                //Exception (termination of method) to handle non-convergent solution
                if (flowCount > flowCountTolerance && isStop == false) {

                  //Set isStop to true
                  isStop = true;

                  //Reset the progressbar to signify end of computation
                  jProgressBar1.setIndeterminate(false);

                  JOptionPane.showMessageDialog(null, "Non-Convergent Flow Solution! Change Time Step or Grid Number to
     Rectify.");
```

```
            flowCount = 0;
            residualValueSum = 0;
            timeDivision = 0;
            flowIterationNumber = 0;
            jif.setTitle(updateTitle);
            for(int i = 0; i < bNum; i++){
              copyE[i] = 0;
            }

            return;
           }

           //Display current iteration status: Time step, number of flow iteration and convergence tolerance
           jif.setTitle(updateTitle + ".  Update: Time Step = " + timeDivision +
                    ".  Flow Count Tolerance = " + flowCount + "." + " Elapsed Time = " +
                    dataFormatUpdateTime2.format( (System.currentTimeMillis() / 1000 - initialTime) / 60)
                    + " min." + " Covg. Toler. = " + dataFormatUpdateTime8.format(residualValueSum));

           //Using "ThomasAlgorithmSolution" method to declare and generate the solution
           double[] flowSolution;

           //Compute Dii for residualValueIntermediate
           Dii = Di(gOfSS, xettaOfSS);

           //Use linear solver based on selected option
           double[] EOneD;
           if(implicitNewtonRaphson.isSelected()){
             EOneD = thomasAlgorithmSolution(dFplusOne(gOfSS, xettaOfSS), dFnone(gOfSS, xettaOfSS), dFminusOne(gOfSS,
xettaOfSS), Dii);
           }
            else{
             EOneD = newtonJacobi(Dii, dFnone(gOfSS, xettaOfSS));
            }

           //Actual flow solution
           flowSolution = flowActualSolution(EOneD, flowCount);

           //Residual computation
           residualValueSum = residual(flowSolution, flowCount);

           //Make a copy of flowSolution for later re-use
           copyE = copy(flowSolution);

           //Plot fractional flow data using "FlowView" object if interactive option is selected
           if(yes.isSelected()){
             fv1.inputdata(ss, copyE, bNum);
             fv1.plotFlowProfile();
           }

          }
          while (flowTolerance < (residualValueSum) && isStop == false); //Do by looping

          if(isStop == false){
           //Copy solution (copyE) into copyETime to be used as initial condition at the next time step
           //or to be used as solution when there is convergent at the final time step.
           copyETime = copy(copyE);

           //Compute distance travelled
           if (timeDivision == 1) {
            normalizedDis = zetta(copyETime, copyETime, copyDisTime,
                           deltaS(), timeDivision);
           }
           else {
            normalizedDis = zetta(copyETime, copyETimeTime, copyDisTime,
                           deltaS(), timeDivision);
           }

           copyDisTime = copy(normalizedDis);

           //Plot saturation-distance using "FlowView" object if interactive option is selected
           if (yes.isSelected()) {
            fv2.inputdata(copyDisTime, ss, bNum);
            fv2.plotSaturationProfile();
```

```
                }

                //Copy solution (copyETime) into copyETimeTime to be used in centered-in-time approximation
                copyETimeTime = copy(copyETime);
              }

            }
            while (timeDivision < (flowIterationNumber) && isStop == false); //Do by looping

          if(isStop == false){
            //Using declared method to generate potential gradients
            norPressPotenOne = potentialGradientOne(copyETime, normalizedDis, deltaS());
            norPressPotenTwo = potentialGradientTwo(copyETime, normalizedDis, deltaS());

            //Calculate MBE
            mbeResult = calculateMBE(normalizedDis);

            //Time at end simulation in seconds
            finalTime = (System.currentTimeMillis() / 1000);

            //Using the printOutput method to print output results
            printOutput(deltaS(), copyETime, normalizedDis, norPressPotenOne,
                    norPressPotenTwo,
                    finalTime, initialTime, flowCount, residualValueSum,
                    norFullTime(), nc(), ng(), mr(), gOfS(),
                    flowIterationNumber,
                    0, cOfS(), fOfS(), timeDivision, mbeResult);
          }

        }
        catch (NumberFormatException nfe) {
          //Exception to catch non-numeric output results
          JOptionPane.showMessageDialog(null, "Missing or Non-Numeric Ouput Value(s).", "Missing or Non-Numeric",
                                JOptionPane.INFORMATION_MESSAGE);
          flowCount = 0;
          residualValueSum = 0;
          timeDivision = 0;
          flowIterationNumber = 0;
          jif.setTitle(updateTitle);
          for(int i = 0; i < bNum; i++){
            copyE[i] = 0;
          }
         return;
        }

        //Reset the progressbar to signify end of computation
        jProgressBar1.setIndeterminate(false);

        if (isStop == false) {
          //Message box to signify end of simulation
          JOptionPane.showMessageDialog(null,"End of Simulation Run. If No Error Message, Check Ouput File for Results.");
          //Reset all relevant static variables
          flowCount = 0;
          residualValueSum = 0;
          timeDivision = 0;
          flowIterationNumber = 0;
          jif.setTitle(updateTitle);
          for(int i = 0; i < bNum; i++){
           copyE[i] = 0;
          }
        }
        //Reset isStop back to true for new run on the thesame frame
        isStop = true;

      } //run method ends here

    }).start(); //thread class ends here


  }else{
    //do nothing
  }//do nothing: ignore
}
```

```
//Save input data to file
void saveInputData_actionPerformed(ActionEvent e) {
  File fileName;
  JFileChooser chooser = new JFileChooser("ics");
  int event = chooser.showSaveDialog(jif);
  fileName = chooser.getSelectedFile();
  if(event == JFileChooser.CANCEL_OPTION) return;

  try{
      //Declare variables/objects
      FileWriter FW = new FileWriter(fileName+".txt");
      PrintWriter PW = new PrintWriter(FW, true);
      int countInt = 0;

      //Set coupling option
      if(viscousCoupling.isSelected()){
        couplingSelection = 'v';
      }
      else if(capillaryCoupling.isSelected()){
        couplingSelection = 'c';
      }
      else if(bothCouplings.isSelected()){
        couplingSelection = 'b';
      }
      else if(noCoupling.isSelected()){
        couplingSelection = 'n';
      }

      //Set R12 selection option
      if(r12.isSelected()){
        r12Selection = '1';
      }
      else if(jRadioButton2.isSelected()){
        r12Selection = '0';
      }

      //Then, save through looping
        for(int i = 1; i < (41+1); i++){
          countInt++;
          //PW.println(Double.parseDouble(jTextFieldArray[countInt].getText()));
          PW.println(jTextFieldArray[countInt].getText());
        }

      //Save R12 selection option
      PW.println(couplingSelection);

      //Save coupling selection option
      PW.println(r12Selection);

      //Close file
      PW.close();
      }

      catch(IOException ioe)
          {
          //Exception to catch wrong output results.
          JOptionPane.showMessageDialog(null,"Wrong Inputs Name! Fill in a Valid Name.");
          }
}

 //Open input data File
 void openInputData_actionPerformed(ActionEvent e) {
  File fileName;
  JFileChooser chooser = new JFileChooser("ics");
  int event = chooser.showOpenDialog(jif);
  fileName = chooser.getSelectedFile();
  if(event == JFileChooser.CANCEL_OPTION) return;

  int countInt = 0;

  //Reset all values to no-value;
  for(int i = 1; i < 41; i = 41){
```

```
 jTextFieldArray[countInt].setText("");
}

 try{
   //Declare variables/objects
   FileReader FR = new FileReader(fileName);
   BufferedReader  BR = new BufferedReader(FR);
   String holdText;

   //Then, read data from file to "jTextFields" through looping
    while((holdText = BR.readLine())!= null){
     countInt++;

     if(countInt <= 41){
      jTextFieldArray[countInt].setText(holdText);

      if(countInt == 23){
       int valueOfSlide = Integer.parseInt(jTextField23.getText());
       jSlider1.setValue(valueOfSlide);
      }

     }

     //Load coupling option
     if(countInt == 42){
      if(holdText.charAt(0) == 'v'){
       viscousCoupling.setSelected(true);
      }
      else if(holdText.charAt(0) == 'c'){
       capillaryCoupling.setSelected(true);
       jTextField29.disable();
       jTextField29.setText("0.0");
      }
      else if(holdText.charAt(0) == 'b'){
       bothCouplings.setSelected(true);
      }
      else if(holdText.charAt(0) == 'n'){
       noCoupling.setSelected(true);
       jTextField29.disable();
       jTextField29.setText("0.0");
      }
     }

     //Load R12 option
     if(countInt == 43){
      if(holdText.charAt(0) == '0'){
       jRadioButton2.setSelected(true);
       jTextField22.disable();
       r12Selection = '0';
      }
      else if(holdText.charAt(0) == 'c'){
       jRadioButton2.setSelected(true);
       jTextField22.enable();
       r12Selection = '1';
       }
     }
    }

   //Close file
   BR.close();

   }

   catch(IOException ioe)
     {
     //Exception to catch wrong output results.
     JOptionPane.showMessageDialog(null,"Wrong Inputs File. Select Another.");
     }
}

//Thomas algorithm method
double [] thomasAlgorithmSolution (double upperElements [], double diagonalElements [],
                                   double lowerElements [], double rhsElements []){

//For an m x n tri-diagonal matrix, number of rows (m) = number of cols n = (matrix size)
int matrixSize = diagonalElements.length;

//Arrays to hold (primary) main computed coefficients, secondary computed coefficients and solutions, x
double A [] = new double [matrixSize];              //Primary
double B [] = new double [matrixSize];              //Primary
double C [] = new double [matrixSize];              //Primary
double D [] = new double [matrixSize];              //Primary
double W [] = new double [matrixSize];              //Secondary
double G [] = new double [matrixSize];              //Secondary
double E [] = new double [matrixSize];              //Solution
double pivot = 0;                           //pivoting value

//Check for zero pivot element

if(diagonalElements[0] == 0){
 //Reset all relevant static variables
 flowCount = 0;
 residualValueSum = 0;
 timeDivision = 0;
 flowIterationNumber = 0;
 jif.setTitle(updateTitle);
 //Set isStop to true to stop the loop
 isStop = true;
 //Reset the progressbar to signify end of computation
 jProgressBar1.setIndeterminate(false);
 //Give a message box to signify exception
 JOptionPane.showMessageDialog(null, "Zero Pivot Error 1 ! Change Time Step or Grid Number to Rectify.");
 for(int i = 0; i < bNum; i++){
  copyE[i] = 0;
 }
}

for(int i = 0; i < matrixSize; i++){
  //Primary
  if(i == 0)
    A[i] = 0;
  else
    A[i] = lowerElements[i - 1];

    B[i] = diagonalElements[i];

  if(i == (matrixSize - 1))
    C[i] = 0;
  else
    C[i] = upperElements[i];

    D[i] = rhsElements[i];

  //Secondary
  if(i == 0){
    W[i] = (C[i] / B[i]);
    G[i] = (D[i] / B[i]);
  }
  else if(i > 0){

    //Pivot condition testing
    pivot = (B[i] - (A[i] * W[i-1]));
    if(pivot == 0){
      //Reset all relevant static variables
      flowCount = 0;
      residualValueSum = 0;
      timeDivision = 0;
      flowIterationNumber = 0;
      jif.setTitle(updateTitle);
      //Set isStop to true to stop the loop
      isStop = true;
      //Reset the progressbar to signify end of computation
      jProgressBar1.setIndeterminate(false);
      //Give a message box to signify exception
      JOptionPane.showMessageDialog(null, "Zero Pivot Error 2 ! Change Time Step or Grid Number to Rectify.");
```

```
     for(int j = 0; i < bNum; i++){
      copyE[j] = 0;
     }
    }

    W[i] = (C[i] / (B[i] - (A[i] * W[i-1])));
    G[i] = ((D[i] - (A[i]*G[i-1])) / (B[i] - (A[i] * W[i-1])));
   }
  }

   //Solution
  for(int i = 0; i < matrixSize; i++){
   int j = (matrixSize - 1) - i;

   if(i == 0){
    E[j] = G[j];
   }
   else if(i > 0){
    E[j] = (G[j] - (W[j] * E[j+1]));
   }
  }
 return E;
}

//A method for printing output results
void printOutput(double deltaS, double [] copyE, double normalizedDis [],
            double norPressPotenOne [], double norPressPotenTwo [],
            double finalTime, double initialTime, int flowCount, double flowResidual,
            double normalizedTime, double nc, double ng, double mr, double gOfS[],
            int timeIntervalNumber, double initTime, double cOfS[],double fOfS[],
            int flowTimestep, double mbe){

 //Instantiating new "DecimalFormat" to be used for formating output results
 DecimalFormat dataFormatOne =  new DecimalFormat("0.0000");
 DecimalFormat dataFormatTwo =  new DecimalFormat("0.0000000000");

 int bNum = copyE.length;
 double normalizedTimee = 0;
 normalizedTimee = ((normalizedTime * timeIntervalNumber) + initTime);
 //Extrapolated values of distance and pressure/potential gradient at zero saturation
 double x = - normalizedDis[1] + 2*normalizedDis[0];
 double p1 = norPressPotenOne[1] - ((normalizedDis[1] - x)/(normalizedDis[1]-normalizedDis[0]))*(norPressPotenOne[1] -
norPressPotenOne[0]);
 double p2 = norPressPotenTwo[1] - ((normalizedDis[1] - x)/(normalizedDis[1]-normalizedDis[0]))*(norPressPotenTwo[1] -
norPressPotenTwo[0]);;

 try
   {
   //Declare variables/objects for saving output results
   ouputNumber = 1 + ouputNumber;
   FileWriter FW = new FileWriter("Output Result" + ouputNumber + ".txt");
   PrintWriter PW = new PrintWriter(FW, true);
   //Then, print(Save) Output results to file
   PW.println("---------------------------------------------------------------------------");
   PW.println("                    Simulation Output Results                      ");
   PW.println("---------------------------------------------------------------------------");

   PW.print("S/No");
   PW.print('\t');
   PW.print("Norm-S");
   PW.print('\t');
   PW.print("fw");
   PW.print('\t');
   PW.print("Pot-Grad1 (Pa/m)");
   PW.print('\t');
   PW.print("Pot-Grad2 (Pa/m)");
   PW.print('\t');
   PW.print("GofS");
   PW.print('\t');
   PW.print("CofS");
   PW.print('\t');
   PW.print("FofS");
   PW.print('\t');
PW.println("Norm-X for Sw & Pot-Grad.");
PW.print("0");
PW.print('\t');
PW.print("0.00");
PW.print('\t');
PW.print("0.0000");
PW.print('\t');
PW.print(String.valueOf(dataFormatOne.format(p1)));
PW.print('\t');
PW.print(String.valueOf(dataFormatOne.format(p2)));
PW.print('\t');
PW.print("0.0000");
PW.print('\t');
PW.print("0.0000");
PW.print('\t');
PW.print("0.0000");
PW.print('\t');
PW.println(String.valueOf(dataFormatOne.format(x)));
for(int i = 0; i < bNum; i++){
  //output variables to hold numerical values
  double v1 = (i+1) * deltaS;
  double v2 = copyE[i];
  double v4One = norPressPotenOne[i];
  double v4Two = norPressPotenTwo[i];
  double v5 = gOfSS[i];
  double v6 = cOfSS[i];
  double v7 = fOfSS[i];
  double v3 = normalizedDis[i];

  //Convert the numeric values to String and format to 4 decimal places
  String v1Text = String.valueOf(dataFormatOne.format(v1));
  String v2Text = String.valueOf(dataFormatOne.format(v2));
  String v4OneText = String.valueOf(dataFormatOne.format(v4One));
  String v4TwoText = String.valueOf(dataFormatOne.format(v4Two));
  String v5Text = String.valueOf(dataFormatOne.format(v5));
  String v6Text = String.valueOf(dataFormatOne.format(v6));
  String v7Text = String.valueOf(dataFormatOne.format(v7));
  String v3Text = String.valueOf(dataFormatOne.format(v3));

  //Do actual writing to file.
  PW.print(i+1);
  PW.print('\t');
  PW.print(v1Text);
  PW.print('\t');
  PW.print(v2Text);
  PW.print('\t');
  PW.print(v4OneText);
  PW.print('\t');
  PW.print(v4TwoText);
  PW.print('\t');
  PW.print(v5Text);
  PW.print('\t');
  PW.print(v6Text);
  PW.print('\t');
  PW.print(v7Text);
  PW.print('\t');
  PW.println(v3Text);

  //Display current iteration status: Time step, number of flow iteration and convergence tolerance
     jif.setTitle(updateTitle + ".  Update: Time Step = " + timeDivision +
     ".  Flow Count Tolerance = " + flowCount + "." + " Elapsed Time = " +
     dataFormatUpdateTime2.format( (System.currentTimeMillis() / 1000 - initialTime) / 60)
     + " min." + " Saving results..... ") ;

}
  //Write out miscellaneous data
  PW.println("---------------------------------------------------------------------------");
  PW.println(                "Miscellaneous Computation Output Data.              ");
  PW.println("---------------------------------------------------------------------------");
  PW.print("Number of flow iteration for convergence: ");
  PW.println(flowCount);
  PW.print("Number of flow time step: ");
  PW.println(flowTimestep);
```

```
            PW.print("Simulation Time (sec): ");
            PW.println(dataFormatOne.format(System.currentTimeMillis() / 1000 - initialTime));
            PW.print("Simulation Time (mins): ");
            PW.println(dataFormatOne.format((System.currentTimeMillis() / 1000 - initialTime)/60));
            PW.print("Simulation Time (hours): ");
            PW.println(dataFormatOne.format((System.currentTimeMillis() / 1000 - initialTime)/3600));
            PW.print("Normalized Time: ");
            PW.println(dataFormatOne.format(normalizedTimee));
            PW.print("Discretized Saturation Domain: ");
            PW.println(dataFormatOne.format((copyE.length) * deltaS));
            PW.print("Max. Normalized Distance: ");
            PW.println(dataFormatOne.format(x));
            PW.print("MBE: ");
            PW.println(dataFormatOne.format(mbe));
            PW.print("Nc: ");
            PW.println(dataFormatTwo.format(nc));
            PW.print("Ng: ");
            PW.println(dataFormatTwo.format(ng));
            PW.print("Mr: ");
            PW.println(dataFormatTwo.format(mr));
            PW.println("------------------------------------------------------------------------");

        //Display current iteration status: nothing happening - end of saving
        jif.setTitle(updateTitle) ;

        //Close file
        PW.close();
      }

    catch(IOException ioe)
      {
        //Exception to catch wrong output results.
        JOptionPane.showMessageDialog(null,"Wrong Outputs! Close the Currently Opened Output Files and Try Running
Simulation Again.");
      }
  }

//ACTUAL COMPUTATION AND SIMULATOR IMPLEMENTATION ENDS HERE

}
```

iv) The Source Codes Listings for FlowView.java

```
package ics;

import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class FlowView extends JPanel {

   public int width = 210, height = 210;       // Default width and height
   public int DOT_SIZE = 2;                    // Size of a graph point
   int bNum;                                   // Size of sat. and flow data
   double [] X = new double[bNum];             // horizontal data
   double [] Y = new double[bNum];             // vertical data

   //Constructor
   public FlowView() {
      setSize(width, height);
  }

   public void inputdata(double [] XX, double [] YY, int bNumm){
     bNum = bNumm;
     X = copy(XX);  //XX converted to percentage/fraction
     Y = copy(YY);  //YX converted to percentage/fraction
   }

   public void plotFlowProfile() {
      Graphics g = this.getGraphics();
      g.setColor(getBackground());             // Clear the drawing area
      g.setClip(0, 0, width, height);
      g.fillRect(0, 0, width, height);
      g.setClip(5, 5, width - 10, height - 10);   // Reset the clip region
      g.translate(width/2, height/2);             // Place origin at middle
      g.setColor(Color.black);
      drawXYAxes();                               // Draw the X and Y axes
      flowProfile();                              // Plot Fractional flow profile
    }

   public void plotSaturationProfile() {
      Graphics g = this.getGraphics();
      g.setColor(getBackground());                // Clear the drawing area
      g.setClip(0, 0, width, height);
      g.fillRect(0, 0, width, height);
      g.setClip(5, 5, width - 10, height - 10);  // Reset the clip region
      g.translate(width/2, height/2);             // Place origin at middle
      g.setColor(Color.black);
      drawXYAxes();                               // Draw the X and Y axes
      saturationProfile();                        // Plot Fractional flow profile
    }

   public void drawXYAxes() {
      Graphics g = this.getGraphics();
      g.translate(width / 2, height/2);
      int hBound = width / 2;                     // Use it to set the bounds
      int vBound = height / 2;
      int tic = width / 100;

      g.drawLine(-hBound,0,hBound,0);             // Draw X-axis
      for (int k = -hBound; k <= hBound; k+=10)
         g.drawLine(k, tic, k, -tic);
         g.drawLine(0, vBound, 0, -vBound);       // Draw Y-axis
      for (int k = -vBound; k <= vBound; k+=10)
         g.drawLine(-tic, k, +tic, k);
   }

   public void flowProfile() {
      Graphics g = this.getGraphics();
      g.translate(width/2, height/2);
      int hBound = width/2;                       // Use it to set the bounds
      g.setColor(Color.red);
      g.fillOval(0, 0, DOT_SIZE, DOT_SIZE);       // Draw the first point

      for (int i = 0; i < bNum; i++) {            // For each pixel on x axis
        g.fillOval((int) X[i], (int) -Y[i], DOT_SIZE, DOT_SIZE); // Draw all other points and reverse y (for cartesian)
      }
   }

   public void saturationProfile() {
     Graphics g = this.getGraphics();
     g.translate(width / 2, height / 2);
     int hBound = width / 2;                      // Use it to set the bounds
     g.setColor(Color.blue);

     for (int i = 0; i < bNum; i++) {             // For each pixel on x axis
       g.fillOval( (int) X[i], (int) - Y[i], DOT_SIZE, DOT_SIZE);  // Draw all other points and reverse y (for cartesian)
     }
   }

   public double [] copy(double [] A){
     int j = A.length;
     double [] E = new double [j];
     for(int i = 0; i < j; i++){
      E[i] = A[i] * 90;
     }
     return E;
   }
}
```

v) The Source Codes Listings for IcsFrame_AboutBox.java

```
package ics;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border.*;
import ics.*;

public class IcsFrame_AboutBox extends JDialog implements ActionListener {
 //Panels
 JPanel panel1 = new JPanel();
 JPanel panel2 = new JPanel();
 JPanel insetsPanel1 = new JPanel();
 JPanel insetsPanel2 = new JPanel();
 JPanel insetsPanel3 = new JPanel();

 //Button
 JButton button1 = new JButton();

 //Labels
 JLabel imageLabel = new JLabel();
 JLabel label1 = new JLabel();
 JLabel label2 = new JLabel();
 JLabel label3 = new JLabel();
 JLabel label4 = new JLabel();

 //Layout formats
 BorderLayout borderLayout1 = new BorderLayout();
 BorderLayout borderLayout2 = new BorderLayout();
 FlowLayout flowLayout1 = new FlowLayout();
 GridLayout gridLayout1 = new GridLayout();

 //Strings
 String product = "Interfacial Coupling Simulator";
 String version = "1.0";
 String copyright = "Copyright (c) 2004, University of Alberta";
 String comments = "Developed by Oluropo Rufus Ayodele";

 //Constructor
 public IcsFrame_AboutBox(Frame parent) {
  super(parent);
  enableEvents(AWTEvent.WINDOW_EVENT_MASK);

  try {
   jbInit();
  }
  catch(Exception e) {
   e.printStackTrace();
  }
  pack();
 }

 //Component initialization
 private void jbInit() throws Exception  {
  //imageLabel.setIcon(new ImageIcon(IcsFrame_AboutBox.class.getResource("[Your Image]")));
  this.setTitle("About");
  setResizable(false);
  panel1.setLayout(borderLayout1);
  panel2.setLayout(borderLayout2);
  insetsPanel1.setLayout(flowLayout1);
  insetsPanel2.setLayout(flowLayout1);
  insetsPanel2.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
  gridLayout1.setRows(4);
  gridLayout1.setColumns(1);
  label1.setText(product);
  label2.setText(version);
  label3.setText(copyright);
  label4.setText(comments);
  insetsPanel3.setLayout(gridLayout1);
  insetsPanel3.setBorder(BorderFactory.createEmptyBorder(10, 60, 10, 10));
  button1.setText("Ok");
  button1.addActionListener(this);
  insetsPanel2.add(imageLabel, null);
  panel2.add(insetsPanel2, BorderLayout.WEST);
  this.getContentPane().add(panel1, null);
  insetsPanel3.add(label1, null);
  insetsPanel3.add(label2, null);
  insetsPanel3.add(label3, null);
```

```
   insetsPanel3.add(label4, null);
   panel2.add(insetsPanel3, BorderLayout.CENTER);
   insetsPanel1.add(button1, null);
   panel1.add(insetsPanel1, BorderLayout.SOUTH);
   panel1.add(panel2, BorderLayout.NORTH);
 }

 //Overridden so we can exit when window is closed
 protected void processWindowEvent(WindowEvent e) {
   if (e.getID() == WindowEvent.WINDOW_CLOSING) {
     cancel();
   }
   super.processWindowEvent(e);
 }

 //Close the dialog
 void cancel() {
   dispose();
 }

 //Close the dialog on a button event
 public void actionPerformed(ActionEvent e) {
   if (e.getSource() == button1) {
     cancel();
   }
 }
}
```

Appendix A2: Listings of the Jacobian Matrix and the RHS of Equations (5.63)

With $i=1,2...4$

The expression, $J * f_i = D_i$, is equivalent to:

$$\begin{pmatrix} (f_1-G_{3/2})^2\chi_{3/2}, & 2\chi_{3/2}(f_1-G_{3/2})[f_2-3f_1+0+G_{3/2}]-1, & 0, & 0 \\ (f_2-G_{5/2})^2\chi_{5/2}, & 2\chi_{5/2}(f_2-G_{5/2})[f_3-3f_2+f_1+G_{5/2}]-1, & (f_2-G_{5/2})^2\chi_{5/2}, & 0 \\ 0, & (f_3-G_{7/2})^2\chi_{7/2}, & 2\chi_{7/2}(f_3-G_{7/2})[f_4-3f_3+f_2+G_{7/2}]-1, & (f_3-G_{7/2})^2\chi_{7/2} \\ 0, & 0, & (f_4-G_{9/2})^2\chi_{9/2}, & 2\chi_{9/2}(f_4-G_{9/2})[0_5-3f_4+f_3+G_{9/2}]-1 \end{pmatrix} * \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} = \begin{pmatrix} (f_1-G_{3/2})^2(f_2-2f_1+0)\chi_{3/2}-f_1+f_1^* \\ (f_2-G_{5/2})^2(f_3-2f_2+f_1)\chi_{5/2}-f_2+f_2^* \\ (f_3-G_{7/2})^2(f_4-2f_3+f_2)\chi_{7/2}-f_3+f_3^* \\ (f_4-G_{9/2})^2(0-2f_4+f_3)\chi_{9/2}-f_4+f_4^* \end{pmatrix}$$

APPENDIX B: A_o and B_o Fitting Coefficients for Equation (6.1) (SSCO - Type II)

Table B1: A_o and B_o fitting coefficients for Equation (6.1) (SSCO - Type II)

Saturation	A_o	B_o
7.77%	0.02295	-0.02022
13.56%	-0.44860	0.10860
25.68%	0.04651	0.60070
34.63%	0.02817	0.19520
46.49%	0.03813	0.19180
63.34%	0.01825	-0.36050
81.21%	0.01087	0.01093
92.89%	-0.04850	0.18210

APPENDIX C: Calibration Data for the Pressure Transducers

Table C1: Calibration data for the pressure transducers

Wetting phase (water)											
Sensor No. 3			Sensor No. 4			Sensor No. 5			Sensor No. 7		
S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value
1	10.00	-9.51744940	1	10.00	-9.59850690	1	10.00	-9.43838120	1	10.00	-10.08491330
2	8.00	-7.60044150	2	8.00	-7.66702800	2	8.00	-7.49345110	2	8.00	-8.14021490
3	6.00	-5.68343350	3	6.00	-5.75241900	3	6.00	-5.61812110	3	6.00	-6.21002900
4	4.00	-3.74909310	4	4.00	-3.75346570	4	4.00	-3.71470690	4	4.00	-4.27023980
5	2.00	-1.83268230	5	2.00	-1.88913390	5	2.00	-1.80493190	5	2.00	-2.31999682
6	0.00	0.09068510	6	0.00	0.05888490	6	0.00	0.08117820	6	0.00	-0.38192320
m	**-1.0405**		**m**	**-1.0352**		**m**	**-1.052**		**m**	**-1.031**	
c	**0.0938**		**c**	**0.0655**		**c**	**0.0948**		**c**	**0.3952**	
Non-wetting phase (light mineral oil)											
Sensor No. 9			Sensor No. 10			Sensor No. 12			Sensor No. 15		
S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value	S/N	Calibrator	Screen Value
1	10.00	-9.62158490	1	10.00	-9.35155490	1	10.00	-9.54246620	1	10.00	-9.00982280
2	8.00	-7.71477030	2	9.99	-9.34791370	2	8.00	-7.64914990	2	9.99	-9.02076630
3	6.00	-5.78718190	3	8.00	-7.43643670	3	6.00	-5.70790100	3	8.00	-7.11846110
4	4.00	-3.84061930	4	6.00	-5.51246600	4	4.00	-3.78329040	4	6.00	-5.20256190
5	2.00	-1.92063210	5	4.00	-3.58849530	5	2.00	-1.85313390	5	4.00	-3.28118820
6	0.00	0.00198910	6	2.00	-1.66431240	6	0.00	0.06593010	6	2.00	-1.35434030
			7	0.01	0.25319960				7	0.00	0.56155920
			8	0.00	0.26530410						
m	**-1.0378**		**m**	**-1.0396**		**m**	**-1.0393**		**m**	**-1.0430**	
c	**0.0040**		**c**	**0.2720**		**c**	**0.0687**		**c**	**0.5815**	

APPENDIX D: Summary of Some USCO and SSCO (Type II) Experimental Data

Table D1: Wetting Phase Injection Rate (Type II - USCO)

Injection Rate (m^3/s)	Time Interval (minutes)	Total Time (minutes)
3.15E-08	3.4660	11.31
3.20E-08	3.4670	14.77
3.29E-08	3.4670	18.24
3.26E-08	3.5170	21.76
2.97E-08	3.4990	25.26
3.34E-08	3.5000	28.76
3.00E-08	3.5010	32.26
3.17E-08	3.4830	35.74
3.12E-08	3.5160	39.26
3.00E-08	3.4670	42.72
3.08E-08	3.4500	46.17
3.08E-08	3.4670	49.64
3.11E-08	3.5160	53.16
3.02E-08	3.5010	56.66
3.03E-08	3.4990	60.16

Table D2: Summary of Measured Data and Other Computed Data (SSCO - Type II)

(A)

S_1 (%)	Average Freq.(MHz.)	S/No.	$q_2(m^3)$	$q_1(m^3)$	dP_2/dx (Pa/m)	P_1/dx (Pa/m)	K_{eff2} (m^2)	K_{eff1} (m^2)	$P_{c\ inlet}$ (Pa)
7.77%	85.046538	1	5.510E-08	0.000E+00	-122,856.91	-	2.2537E-11	0.0000E+00	3,201.93
13.56%	84.987085	2	4.295E-08	5.883E-08	-115,239.98	-98,320.50	1.8730E-11	1.1967E-12	2,432.68
25.68%	84.979352	3	2.118E-08	3.161E-08	-86,875.78	-83,512.64	1.2253E-11	7.5704E-13	1,658.19
34.63%	84.958641	4	6.343E-09	3.565E-08	-74,011.39	-42,460.21	4.3066E-12	1.6794E-12	956.32
46.49%	84.943687	5	3.343E-09	4.590E-08	-64,649.84	-64,534.93	2.5984E-12	1.4225E-12	645.14
63.34%	84.934656	6	2.343E-09	5.510E-08	-63,565.83	-42,490.86	1.8522E-12	2.5935E-12	435.23
81.21%	84.871784	7	1.143E-09	6.300E-08	-35,285.83	-14,364.08	1.6277E-12	8.7719E-12	134.56
92.89%	84.850492	8	0.000E+00	8.565E-08	-	-13,908.26	0.0000E+00	1.2317E-11	0.00

(B)

S_1 (%)	K_{r2} (fitted)	K_{r1} (fitted)	S	1-S	S_1 (%)
7.77%	1.0000	0.0000	0.0000	1.00	7.77%
13.56%	0.7893	0.0095	0.0680	0.93	13.56%
25.68%	0.4545	0.0553	0.2104	0.79	25.68%
34.63%	0.2839	0.1077	0.3156	0.68	34.63%
46.49%	0.1354	0.2022	0.4549	0.55	46.49%
63.34%	0.0321	0.3961	0.6528	0.35	63.34%
81.21%	0.0019	0.7081	0.8628	0.14	81.21%
92.89%	0.0000	1.0000	1.0000	0.00	92.89%

APPENDIX E: Frequency and Pressure Data of USCO (Type I) Experiment

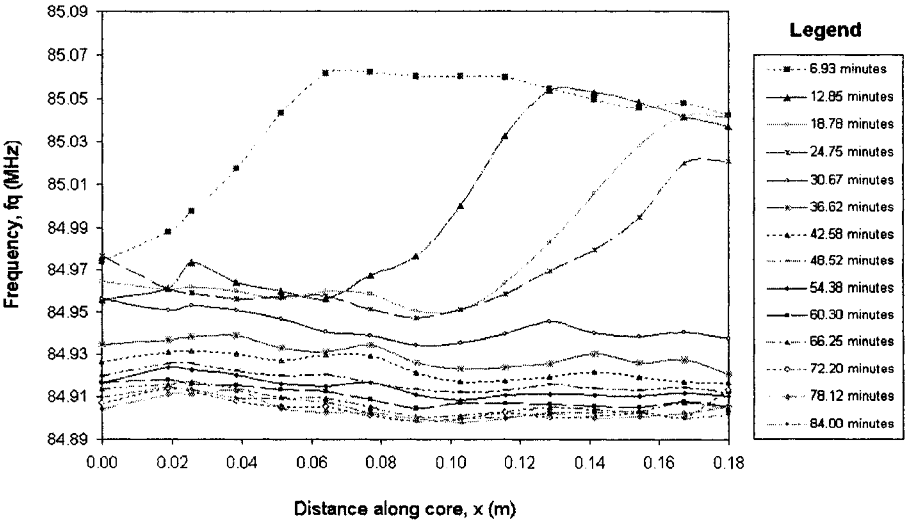


Figure E1 : Raw frequency measurements along the core
(Data Group B: Type I - USCO)

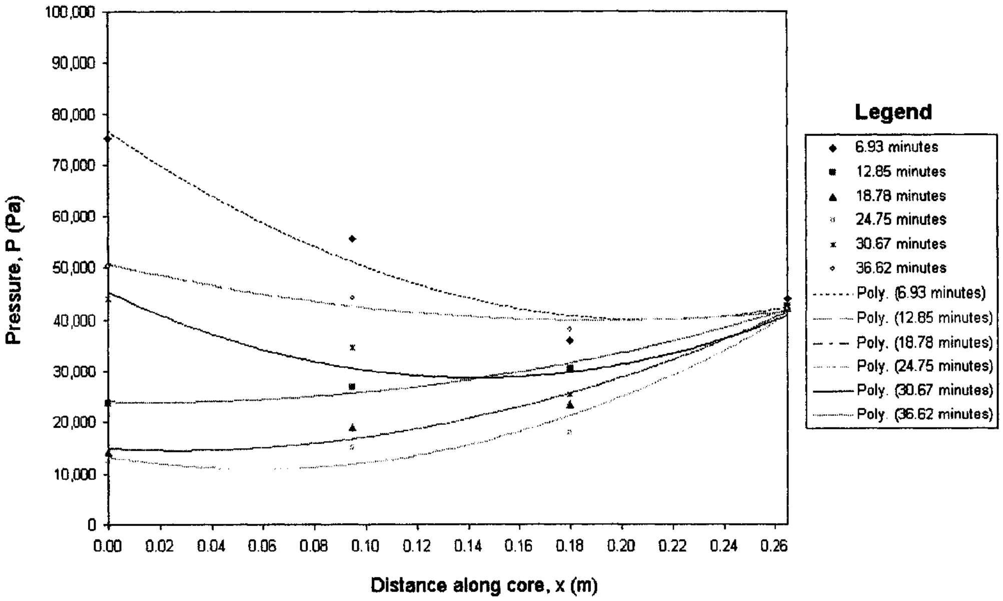


Figure E2 : Wetting phase pressure measurements along the core (Type I - USCO)

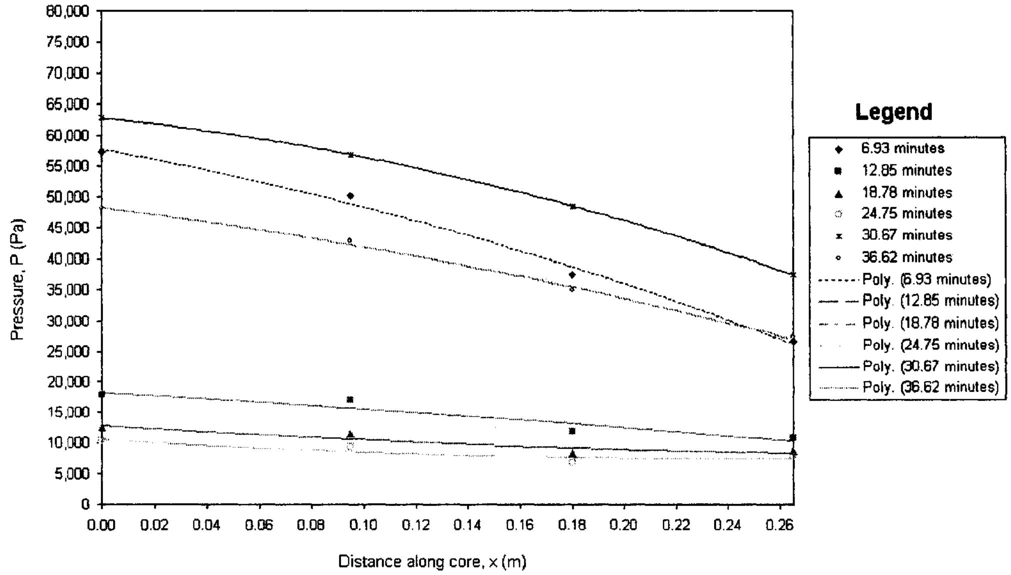


Figure E3 : Nonwetting phase pressure measurements along the core (Type I - USCO)