

PREDICTION OF MULTIPHASE FLOW PROPERTIES FROM
NUCLEAR MAGNETIC RESONANCE IMAGING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

TÜRKER KARAMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
PETROLEUM AND NATURAL GAS ENGINEERING

FEBRUARY 2009

Approval of the thesis:

**PREDICTION OF MULTIPHASE FLOW PROPERTIES FROM
NUCLEAR MAGNETIC RESONANCE IMAGING**

submitted by **TÜRKER KARAMAN** in partial fulfillment of the requirements for
the degree of **Doctor of Philosophy in Petroleum and Natural Gas Engineering**
Department, Middle East Technical University by,

Prof. Dr. Canan ÖZGEN _____
Dean, Graduate School of Natural and Applied Sciences

Prof. Dr. Mahmut PARLAKTUNA _____
Head of Department, Petroleum and Natural Gas Eng.

Prof. Dr. Serhat AKIN _____
Supervisor, Petroleum and Natural Gas Eng. Dept., METU

Prof. Dr. M.R. Birol DEMİRAL _____
Co-Supervisor, Geosciences and Petroleum Eng. Dept., UTP

Examining Committee Members:

Prof. Dr. M.R. Birol DEMİRAL _____
Geosciences and Petroleum Eng. Dept., UTP

Prof. Dr. Serhat AKIN _____
Petroleum and Natural Gas Eng. Dept., METU

Prof. Dr. Mahmut PARLAKTUNA _____
Petroleum and Natural Gas Eng. Dept., METU

Assoc. Prof. Dr. Bora ROJAY _____
Geological Engineering Dept., METU

Asst. Prof. Dr. Evren ÖZBAYOĞLU _____
Petroleum and Natural Gas Eng. Dept., METU

Date: 19 February 2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname : Türker KARAMAN

Signature :

ABSTRACT

PREDICTION OF MULTIPHASE FLOW PROPERTIES FROM NUCLEAR MAGNETIC RESONANCE IMAGING

KARAMAN, Türker

Ph.D., Department of Petroleum and Natural Gas Engineering

Supervisor: Prof. Dr. Serhat AKIN

Co-Supervisor: Prof. Dr. M.R. Birol DEMİRAL

February 2009, 151 pages

In this study a hybrid Pore Network (PN) model that simulates two-phase (water-oil) drainage and imbibition mechanisms is developed. The developed model produces Nuclear Magnetic Resonance (NMR) T2 relaxation times using correlations available in the literature. The developed PN was calibrated using experimental relative permeability data obtained for Berea Sandstone, Kuzey Marmara Limestone, Yeniköy Dolostone and Dolomitic

Limestone core plugs. Pore network body and throat parameters were obtained from serial computerized tomography scans and thin section images. It was observed that pore body and throat sizes were not statistically correlated. It was also observed that the developed PN model can be used to model different displacement mechanisms.

By using the synthetic data obtained from PN model, an Artificial Neural Network (ANN) model was developed and tested. It has been observed that the developed ANN tool can be used to estimate oil – water relative permeability data very well (with less than 0.05 mean square error) given a T2 signal. It was finally concluded that the developed tools can be used to obtain multiphase flow functions directly from an NMR well log such as Combinable Magnetic Resonance (CMR).

Keywords: Pore Networks, Two Phase Relative Permeability, Artificial Neural Networks, Nuclear Magnetic Resonance

ÖZ

NÜKLEER MAGNETİK REZONANS GÖRÜNTÜLEME İLE ÇOK FAZLI AKIŞ ÖZELLİKLERİNİN TAYİNİ

KARAMAN, Türker

Doktora, Petrol ve Doğal Gaz Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Serhat AKIN

Ortak Tez Yöneticisi: Prof. Dr. M.R. Birol DEMİRAL

Şubat 2009, 151 sayfa

Bu çalışmada iki fazlı (su-petrol) drenaj ve imbibisyon mekanizmalarını tasvirleyen hibrit bir Gözenek Ağı (GA) modeli geliştirilmiştir. Geliştirilen model literatürde yeralan bağıntılar kullanarak Nükleer Magnetik Rezonans (NMR) T2 gevşeme zamanlarını üretmektedir. Geliştirilen GA Berea Kumtaşı, Kuzey Marmara Kireçtaşı, Yeniköy Dolomit ve Dolomitli Kireçtaşı tapaları

kullanılarak elde edilen deneysel görelı geçirgenlik verileri ile kalibre edilmiştir. GA gözenek ve bağlantı parametreleri bilgisayarlı tomografi taramaları ve ince kesit görüntülerinden elde edilmiştir. Gözenek ve bağlantı boyutlarının istatistiksel olarak ilişkisiz olduđu gözlemlenmiştir. Ayrıca, geliştirilen GA modelinin farklı öteleme mekanizmalarını modellemek üzere kullanılabileceđi gözlenmiştir.

GA modelinden elde edilen sentetik veriler kullanılarak bir Yapay Sinir Ađı (YSA) modeli geliştirilmiş ve test edilmiştir. Geliştirilen YSA modelinin verilen T2 gevşeme sinyali için petrol – su görelı geçirgenlik verilerini yeterince iyi bir şekilde tahmin edebildiđi görölmüştür (ortalama karesel hata 0.05'ten daha küçüktür). Sonuç olarak, geliştirilen modellerin Kombine Magnetik Rezonans (CMR) gibi bir NMR kuyu logu direk olarak kullanılarak çok fazlı akış fonksiyonalarını elde etmede kullanılabileceđi sonucuna ulaşılmıştır.

Anahtar Kelimeler: Gözenek Ađları, İki Fazlı Görelı Geçirgenlik, Yapay Sinir Ađları, Nükleer Magnetik Rezonans

To My Family

ACKNOWLEDGEMENTS

I would like to express all my respect and sincere gratitude to my supervisor Prof. Dr. Serhat Akın for his continuous support, guidance and encouragement during the developments of this study.

I am also grateful to my cosupervisor Prof. Dr. M.R. Birol Demiral for his help, ideas and suggestions during the course of this research.

I send my sincere gratitude to Prof. Dr. Mahmut Parlaktuna, Assoc. Prof. Dr. Bora Rojay and Asst. Prof. Dr. Evren Özbayoğlu for giving their time and attention to serve in my committee.

I would like to thank my parents for their support giving me the strength to overcome all difficulties that I am facing in my life.

I would like to express my special thanks to my wife, Nesteren, for being on my side during this study. Her invaluable motivation and trust in me was extremely helpful in completion of this study.

And to my sons, Yiğit Batuhan and Yağız Baturalp, I want to express my apologies for my frequent absences.

TABLE OF CONTENTS

ABSTRACT	iv
ÖZ.....	vi
ACKNOWLEDGEMENTS.....	ix
TABLE OF CONTENTS.....	x
LIST OF TABLES	xv
LIST OF FIGURES	xvi
NOMENCLATURE	xxi
CHAPTER	
1. INTRODUCTION.....	1
2. LITERATURE SURVEY	3
2.1. Pore Network Modelling.....	3
2.1.1. Why Use of Pore Network Models is Necessary?	3
2.1.2 Brief Historical Remarks	4
2.1.3. Pore Network Model Architecture	7

2.1.3.1. The Common Pore Network Model	7
2.1.3.2. A More Advanced Model for Sandstones ...	11
2.1.4. Basics of Network Models	15
2.1.5. Running the Pore Network Model	18
2.1.5.1. Capillary and Gravitational Forces	19
2.1.5.2. Conductivity	21
2.1.5.3. Material Balance	22
2.1.5.4. Implementation for Solving	22
2.2. Artificial Neural Networks	23
2.2.1. What is a Neural Network?	24
2.2.2. Inspiration from the Neuroscience	24
2.2.3. Neural Computational Models	26
2.2.4. Learning Process	27
2.2.4.1. Unsupervised (or Self-organization)	
Learning	28
2.2.4.2. Supervised (or Associative) Learning	29
2.3. NMR Relaxation Mechanisms	29

2.3.1. NMR Relaxation Mechanisms	
of Fluids in Pores.....	29
2.3.1.1. Bulk Relaxation.....	32
2.3.1.2. Surface Relaxation	33
2.3.1.3. Diffusion-induced Relaxation	34
2.3.2. Multi-exponential Decay.....	36
2.3.3. Echo-fit for T2 distribution	40
2.3.4. Determination of Bulk Volume of	
Irreducible Water (BVI)	40
2.3.5. Magnetic Resonance Imaging Log (MRIL)	
Permeability Model	42
2.3.5.1. The Free-Fluid (or Coates) model	42
2.3.5.2. The Mean T2 (or SDR) Model	42
2.3.6. MRIL Porosity Model	43
3. STATEMENT OF THE PROBLEM.....	45
4. METHODOLOGY	47
5. IMPLEMENTATION TO THE PORE NETWORK	
MODELLING	50

5.1. CT and Thin Section Image Processing	50
5.2. NMR Subroutine	64
5.3. Geological Investigation on Core Plugs	65
6. IMPELMANTATION TO THE NEURAL NETWORK	
MODELLING	66
6.1. Calculation of Error and Finding Global	
Minimum at Error Space	67
6.2. Sensitivity Study on ANN Model While Learning	
With Data of Water-wet Systems	68
6.3. Enlargement of the Span of the Training Data	75
7. CONCLUSIONS	79
8. RECOMMENDATIONS	82
REFERENCES	83
APPENDICIES	
A. SOURCE CODE OF PORE NETWORK MODELLING	93
B. SOURCE CODE OF THE ARTIFICIAL NEURAL	
NETWORK PROGRAM	125
C. NEURAL NETWORK CONSTRUCTION PHILOSOPHY	133

C.1. Backpropagation Algorithm	133
C.2. Neuron Activation Functions.....	135
C.3. Learning Rate and Momentum Term.....	135
C.4. Exclusive OR (XOR) Function.....	135
D. CT IMAGES.....	139
E. THIN SECTION IMAGES - POROSITY and PORE SIZE DISTRIBUTIONS.....	145
CURRICULUM VITAE	150

LIST OF TABLES

TABLES

5-1 Core Plug Information	50
5-2 Results of t-test on Thin Section #1 and #2 of Core Plug 3-3	55
5-3 Results of t-test on Thin Section #3 and #4 of Core Plug 3-3	55
5-4 Results of F-test on Thin Section #3 and #4 of Core Plug 3-3	56
5-5 Pore Network Model Radius Distribution	
Parameters of Core Plugs	58
5-6 Geological Investigation Results on Core Plugs	65
C-1 The Famous XOR Function.....	136
C-2 Testing Data and the Corresponding Output for 0.001 error.....	137
C-3 Testing Data and the Corresponding Output for 0.0001 error.....	137

LIST OF FIGURES

FIGURES

2-1	Network model of a porous medium.....	8
2-2	A pore representation	9
2-3	3D representation of a porous medium	10
2-4	Sedimentation model for sandstone	12
2-5	The simulated linear compaction process in sandstone for compaction factors of 0.0, 0.1, 0.2 and 0.3.....	13
2-6	The skeleton view for a generated sandstone.....	14
2-7	The ball (pore-body) and stick (pore-throat) representation of a 3-D network model	15
2-8	Imbibition in pores	19
2-9	Schematic drawing of a typical neuron	25
2-10	Illustration of multi-exponential decay character of a porous medium.....	38

2-11	Echo-fit for T2 Distribution	40
2-12	Cut-off BVI Model.....	41
2-13	Spectral BVI Model	41
2-14	NMR porosity model for a water-wet formation.....	43
4-1	Flow chart of the methodology	48
5-1	CT images of Core Plug 3-3 (Limestone)	51
5-2	Porosity Distribution in Core Plug 3-3.....	52
5-3	3x3 imaging of first thin section of Core Plug 3-3.....	53
5-4	Representation of Pore and Particles for First Thin Section of Core Plug 3-3	54
5-5	Porosity Distribution for Plug 3-3 from Thin Section Imaging	56
5-6	Frequency & Cumulative % vs. Radius (Plug 3-3).....	57
5-7	Calculated Capillary Pressures for Systems Having Different Contact Angles.....	60
5-8	Relative Permeabilities for Plug 92-2 (before modification in calculations).....	61
5-9	Relative Permeabilities for Plug 100-4 (before modification in calculations).....	62

5-10	Relative Permeabilities for Plug 92-2 (after modification in calculations).....	63
5-11	Relative Permeabilities for Plug 100-4 (after modification in calculations).....	63
6-1	Generated Data Set by Pore Network Model for Part # 1	68
6-2	Schematic diagram of two-phase relative permeability model (Stage#1)	69
6-3	Schematic diagram of two-phase relative permeability model (Stage#2)	70
6-4	Schematic diagram of two-phase relative permeability model (Stage#3)	71
6-5	Schematic diagram of two-phase relative permeability model (Stage#4)	72
6-6	Comparison of Oil Relative Permeabilities Obtained From PN and ANN (Contact Angle = 0°)	73
6-7	Comparison of Water Relative Permeabilities Obtained From PN and ANN (Contact Angle = 0°)	74
6-8	Generated Data Set by Pore Network Model for Part # 2	75

6-9	Comparison of Oil Relative Permeabilities Obtained	
	From PN and ANN (Contact Angle = 90°)	76
6-10	Comparison of Water Relative Permeabilities Obtained	
	From PN and ANN (Contact Angle = 90°)	77
6-11	Comparison of Oil Relative Permeabilities Obtained	
	From PN and ANN (Contact Angle = 180°)	78
6-12	Comparison of Water Relative Permeabilities Obtained	
	From PN and ANN (Contact Angle = 180°)	78
C-1	A 2:4:1 network.....	136
D-1	CT images of Plug 4-1 (Sandstone)	140
D-2	CT images of Plug 92-2 (Dolostone)	140
D-3	CT images of Plug 100-4 (Dolomitic Limestone).....	141
D-4	CT images of Plug 91-5 (Dolostone)	141
D-5	CT images of Plug 83-6 (Dolomitic Limestone).....	142
D-6	Porosity Distribution in Plug 92-2 (Dolostone)	142
D-7	Porosity Distribution in Plug 91-5 (Dolostone)	143
D-8	Porosity Distribution in Plug 100-4 (Dolomitic Limestone).....	143
D-9	Porosity Distribution in Plug 83-6 (Dolomitic Limestone).....	144
D-10	Porosity Distribution in Plug 4-1 (Sandstone)	144

E-1	3x3 imaging of third thin section of Plug 4-1	145
E-2	3x3 imaging of fifth thin section of Plug 92-2.....	146
E-3	3x3 imaging of seventh thin section of Plug 100-4.....	146
E-4	Porosity Distribution for Plug 4-1 (sandstone) from Thin Section Imaging	147
E-5	Porosity Distribution for Plug 92-2 (Dolostone) from Thin Section Imaging	147
E-6	Porosity Distribution for Plug 100-4 (Limestone) from Thin Section Imaging	148
E-7	Frequency & Cumulative % vs Radius (Plug 4-1).....	148
E-8	Frequency & Cumulative % vs Radius (Plug 92-2).....	149
E-9	Frequency & Cumulative % vs Radius (Plug 100-4).....	149

NOMENCLATURE

α	=	corner half angle = 30° (for equilateral triangles)
A	=	cross-sectional area of a pore
β	=	dimensionless resistance factor
∇P	=	pressure gradient
γ_{ij}	=	interfacial tension between phase i and j
g	=	conductance
k	=	relative permeability
l	=	radius
L	=	length of a pore
μ	=	viscosity
P	=	pressure
P_c	=	capillary pressure
Q	=	volumetric flowrate
R	=	inscribed radius
S	=	saturation
V	=	volume of a pore
P_o^D	=	oil pressure for oil to enter into pore during primary drainage
P_o^f	=	oil pressure at the end of primary drainage/aging for all pores
P_w^*	=	water pressure during primary drainage (constant throughout process)
$P_{c,ow}^D$	=	oil/water capillary pressure for oil to enter into pore during primary drainage

Subscripts

A	=	advancing
Crit	=	critical
C	=	corner
Ct	=	center
eff	=	effective
i	=	phase index (o or w)
j	=	pore index
o	=	oil phase
R	=	receding
si	=	single phase i (when i occupies the pore completely)
T	=	total
w	=	water phase

Superscripts

D	=	primary drainage
*	=	at the start of process
f	=	at the end of process

CHAPTER 1

INTRODUCTION

Reservoir Engineers do need to know what is going on inside a reservoir. In order to understand this there are some tools like open hole and production logs, well tests, simulators, multiphase flow meters and core analysis. But all of these analysis tools are for macroscopic evaluation. Given the chaotic movements of the fluids in a small core plug it can even be said that core analysis is macroscopic.

A major problem in performing realistic field reservoir simulation studies is scaling the laboratory core measurements to full field-scale. Relative permeability, residual oil saturation and capillary pressure measurements are usually influenced by reservoir heterogeneity that occurs at all scales in the reservoir; from the pore-scale to the full field-scale. These properties are averaged representations of properties that occur at pore scale and there is always a possibility to estimate them by generating the pore scale activity virtually.

This idea arisen the pore network modeling philosophy. Estimation of the relative permeabilities by using pore based networking models has become popular because of their advanced applicability for calculations of the relative permeability values for individual saturation values. The results

obtained from a pore network model are very similar to those found from laboratory experiments by using the real core samples.

Second focus of this study is Nuclear Magnetic Resonance (NMR) calculations. NMR well logging has been commercially available since the early 1990s. More recently, NMR has been used for real-time analysis of bottom-hole sampling of reservoir fluids. Interpretation of these measurements require an understanding of the relationship between NMR dependent parameters, hydrogen index, T1 and T2 relaxation time, and diffusion coefficient with the composition, density, pressure, temperature, and viscosity of the fluids. With this understanding, it is possible to estimate the formation porosity and permeability, irreducible water saturation, saturation of water, oil, and gas, and the oil viscosity using an NMR log.

In this study a pore network model that generates NMR T2 relaxation times along with the corresponding oil – water displacement relative permeability was coupled with a neural network model. It is intended to predict two phase relative permeability values for different reservoir systems with differing wettabilities. The aims of this coupling decision are to increase the applicability and practicality of the pore network models and also to construct more reliable, flexible, dynamic and adjustable two phase relative permeability predictor that inputs some basic properties like fluid viscosities, formation temperature, surface relaxivity (changes with the rock type) and T2 relaxation times of different rock-fluid systems.

CHAPTER 2

LITERATURE SURVEY

2.1. Pore Network Modeling

2.1.1. Why Use of Pore Network Models is Necessary?

One of the major problems in performing realistic field reservoir simulation studies is relating laboratory core measurements of relative permeability and residual oil saturation to the field scale. This is because relative permeability and residual oil saturation are strongly influenced by reservoir heterogeneity and heterogeneity occurs at all scales in the reservoir from the pore-scale to the full field-scale (Kamath et. al., 1996; Paterson et. al., 1996; Narayanan and Deans, 1988).

Relative permeability measurements, either by steady state or unsteady state methods, are time consuming, expensive, and often difficult to interpret. As a result, too few measurements are usually performed and numerous uncertainties may be associated with the measurements. This prohibits assigning unique relative permeability functions to different architectural units in the reservoir (i.e., channels, crevasse splays, wash-over fans, etc.) and limits

the ability of reservoir simulators to accurately predict oil recovery (Nilsen et. al., 1996; Roberts et. al., 1996).

Relative permeabilities and capillary pressures are averaged transport properties, which represent the physical process occurring on the pore-scale. On the pore-scale, the displacement of one fluid by another is controlled by interfacial tension, viscous forces, rock-fluid interactions, and the geometry of the pore space. In principle, it should therefore be possible to determine relative permeabilities and capillary pressure by appropriately averaging the equations describing the physical processes occurring on the microscopic or pore-scale. This approach requires a detailed understanding of the displacement mechanisms on the pore-scale and a complete description of the morphology of the pore space (Dixit et. al., 1996; Nilsen et. al., 1996; Narayanan and Deans, 1988).

Although the literature is populated with this subject, determination of relative permeabilities is not the only use of pore network models. Aside from uses in chemical engineering, hydrology, physics, pore network models are also used in petroleum engineering to study underground gas storage, steam injection, foam formation, and formation damage (Davies and Vessell, 1995; Kharabaf and Yortsos, 1996; Wenrong et. al., 1996; Satik and Yortsos, 1995; Billiotte et. al., 1990).

2.1.2. Brief Historical Remarks

The below is summarized from a section in the “U.S. National Report to IUGG, 1991-1994” (Celia and Reeves, 1995). The summary involves many other references, which are not listed in the references section of this study. Still, all those references are kept; as, the section by Celia and Reeves (1995) proves to be a comprehensive research on pore network models; and,

accordingly, the below summary is probably the summary of one of the most comprehensive texts about the history of pore network models:

Historically, capillary-dominated multi-phase porous media systems have been modeled by analogy to fluid behavior in capillary tubes, where the pore space of a porous solid is idealized as a collection of capillaries. A highly incomplete list of references of historical significance include Washburn (1921), who studied dynamic invasion of a fluid into a capillary, Carmen (1937), who used the concept of hydraulic radius to define equilibrium positions of fluid-fluid interfaces in tubes of difference cross sections, Haines (1930), who examined the motion of unstable interfaces, and Fatt (1956a, 1956b, 1956c), who introduced the idea network models and whose work forms the basis for much of the recent research described herein. An excellent review of pore-scale models through the mid-1970's is provided by van Brakel (1975), who presents a comprehensive and critical review with a strong sense of history.

While use of bundle-of-tubes (The good old mercury injection...) allows simple analytical formulas to be derived for properties such as relative permeability, such models miss a fundamentally important topological characteristic of porous solids, namely interconnectedness of the pore space. This failing has motivated the development of models of the pore space that arrange capillary tubes in two- or three-dimensional networks. The movement of fluids, and the interfaces that separate different fluid phases, through such a network of capillary tubes, forms the basis of pore-scale network models. The pioneering works in these types of network models were that of Fatt (1956), who used two-dimensional networks of capillary tubes with randomly distributed radii to model the pore space. Since Fatt's 1956 publications, there have been many extensions of the network modeling approach. While Fatt used

volumeless junctions in the networks, most models now use pore-space descriptions that include junctions characterized by an effective radius, where the junction is referred to as a pore body or a 'site' of the lattice. The pore bodies are meant to correspond to the larger void spaces found in natural porous media. The narrow openings that connect adjacent pore bodies are modeled by the capillary tubes of the network model, which are called pore throats or 'bonds' of the lattice. Pore bodies are usually represented by spheres whose size distribution is representative of pore bodies found in the particular porous medium of interest, while the pore throats are usually represented by cylinders or more general conical shapes with an analogous size distribution that is characteristic of the narrow openings that connect the pore bodies. Fenwick and Blunt (1997) and Karaman and Demiral (2003a, June 2004) have used equilateral triangular pores in order to have residual saturations at the edges. Connectedness is usually characterized by coordination number, which corresponds to the number of bonds that meet at a site.

Since Fatt's original publication, a large number of publications have appeared related to network modeling and pore-scale displacement processes. Most of these have appeared in the chemical engineering, petroleum engineering, and physics literature. These include fundamental work in computational methods; theoretical developments, usually involving aspects of percolation theory; and experimental studies, often involving etched-plate micromodels. An excellent reference for details of various pore-scale processes, including detailed descriptions of many aspects of network models, is the classic text of Dullien (1991).

Closely allied with computational network models is the concept of percolation theory. Generally credited as being introduced in 1957 by Broadbent and Hammersley, percolation theory provides a mathematical

framework to analyze deterministic motions in a random medium. In the case of fluid-fluid displacement in porous solids, the deterministic rule is the Laplace equation for capillary displacement rule (Dullien, 1991), while the randomness comes from the random tube radii defined on the lattice that represents the pore space. This is a powerful tool that can shed light on various aspects of multiphase displacement problems. The general topic of percolation theory, with specific application to hydrology, has been reviewed recently by Berkowitz and Balberg (1992). The basic concepts of percolation theory are not reviewed herein, and recent developments in percolation theory are not included in this review except to the extent that the computational network model results discussed below involve components of percolation theory.

2.1.3. Pore Network Model Architecture

This part, first try to reveal the basics behind the common pore network model architecture and then, another common model developed to characterize sandstones (but, that can be used to simulate any kind of rock) will be discussed.

2.1.3.1. The Common Pore Network Model

To understand better, the common pore network model that is widely used in the literature for almost every kind of rocks, consider the below figure (Figure 2-1) taken from Koplik and Lasseter (1985):

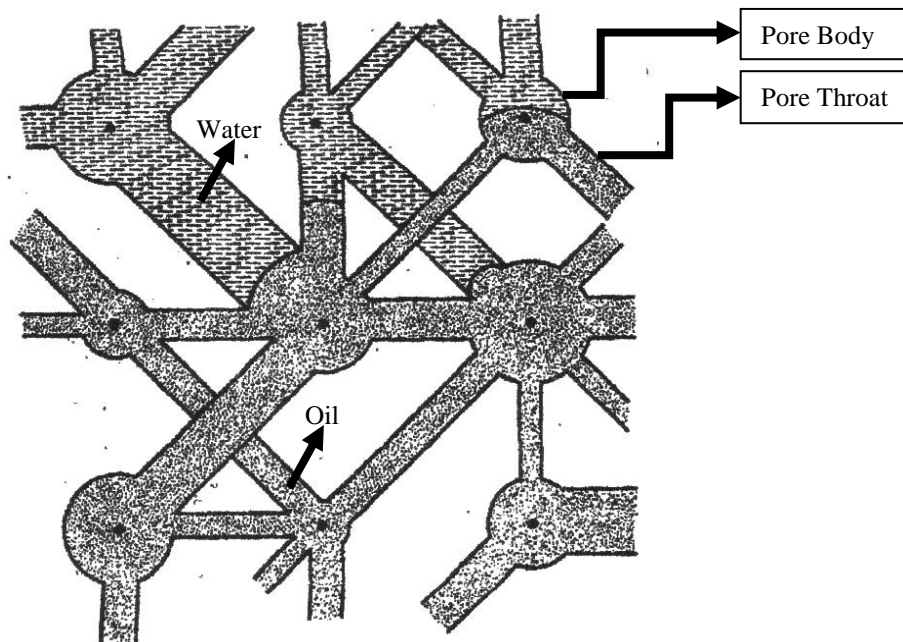


Figure 2-1: Network model of a porous medium. (after Koplik & Lasseter, 1985)

This figure is self explaining: By assigning circles to represent pores, and by assigning tubes to represent the pore connections (or, ‘small’ pores in a way) it was managed to generate a simplified version of the actual pore structure. A more detailed research of this figure will reveal another fact about the today’s pore network models: Although looking like a complex model, the above structure is a simple one with pore (circle) centers having constant distance to each other. Realizing this simplified approach, one can further understand this 2D pore network model by only extensive examination of single pores. Consider the below figure:

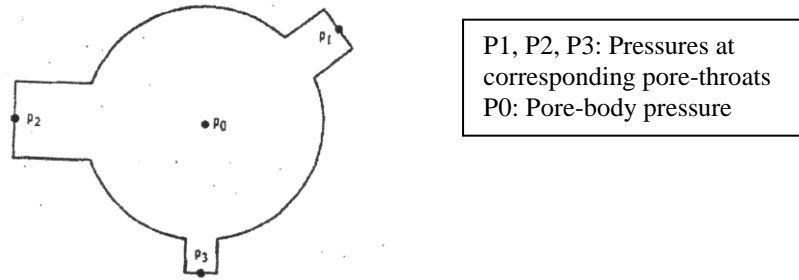


Figure 2-2: A pore representation. (after Koplik & Lasseter, 1985)

This most basic element of the pore network could have the following parameters (Note that, the determination of these parameters will be explained in the following paragraphs):

1. **Radius;** the radius of the circle (2D) or sphere (3D).
2. **Shape Factor;** at low wetting phase saturations, the continuity and mobility of the wetting phase are mainly controlled by the amount of wetting fluid retained in the corners of pores with angular cross-section. The pore shape factor, together with capillary pressure, determine the amount of wetting fluid retained in the corners of non-wetting fluid filled pores. The hydraulic conductivity of the wetting phase is, therefore, strongly influenced by the magnitude of the shape factor (Nilsen et. al. 1996). This is why we need to have a shape factor for our ‘artificial’ pores. Still, the actual representation of the shape factor is limited; and only few studies reviewed in this report account for shape factors (Nilsen et. al. 1996; Lenormand et. al, 1984).
3. **Coordination Number;** the average number of pore throats connected to a pore body. It is assumed that, if one manages to correctly identify this coordination number, then the network model generated

accordingly will, at least, have some properties similar to the original pore structure.

4. **Aspect Ratio**; the ratio of pore body radius to pore throat radii.

5. **Pore Throat Radius**; the radius of the connections connecting pore bodies. Note that, there are examples in the literature that demonstrates conic pore throats instead of tubes; that is, the entry of a pore throat to a specific pore has different radius to the other end of that throat.

These features of a pore are common to almost all pore network models studied in the literature. When combined to a 3D structure, this pore model adapts itself accordingly; and, has several similarities to the 2D model (Figure 2-3):

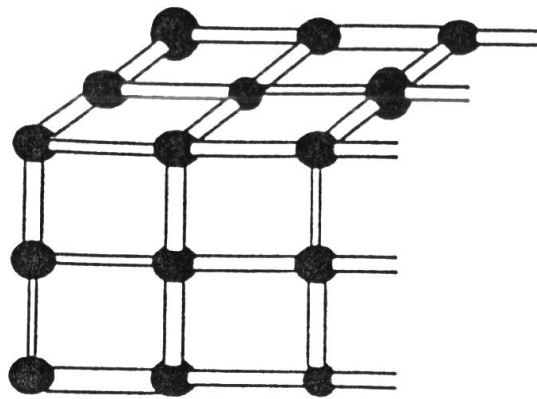


Figure 2-3: 3D representation of a porous medium.

In this pore model, as obviously seen from the figure, the locations of pore bodies are predetermined. They are located at the intersection points of 3D grid, which is designed to cover the entire ‘artificial’ core sample. The problem

of implementing this model, then, reduces to the decision of pore body radii and throat radii. There are several techniques in the literature to determine these values like the Rayleigh distribution, stochastic approach and Laplace equation (Bakke and Oren, 1996; Kamath et. al., 1996; Xu et. al., 1997; Koplik and Lasseter, 1985; Ioannidis et. al., 1995; Kharabaf and Yortsos, 1996; Wenrong et. al., 1996).

Yet, again all of the above available techniques make use of some advanced techniques to acquire information about the porous medium. Although techniques such as serial-sectioning and micro-CT are available, information about the pore structure of porous rocks is usually obtained from image analysis of 2D thin section images of rocks (Nilsen et. al., 1996) and from mercury injection capillary pressure curves (Bundle of tubes approach). Data coming from these are organized in plots, or functions allowing a random distribution for the network to be implemented. Then, using those plots or functions radius of each pore body and pore throat is determined.

2.1.3.2. A More Advanced Model for Sandstones

Although specifically designed to simulate sandstones (Bakke and Oren, 1996; Nielsen et. al., 1996), the below model could also be adapted to any kind of reservoir rock (In fact, a study by Roberts et. al. (1996) came up with such a model of pore structure that is similar to the ‘sandstone model’.).

Sandstones are the end products of a series of complex geological processes, which, starts with erosion of a quartz bearing rock, followed by a grain transport via air, water or ice, and finally, sedimentation of grains in a basin. The deposited grains may be reworked by one or several cycles of erosion, transport, and sedimentation. The sandstone generation process is completed by compaction and various diagenetic processes. Information

regarding the last sedimentation cycle and subsequent compaction and diagenesis may be interpreted from thin section analysis.

The exact, complex methodology used to recreate each of these steps in the generation of sandstone (Sedimentation, Compaction, and Diagenesis) is beyond this reports coverage. Yet, the reader should note that the recreation, especially the sedimentation, involves random procedures (e.g., the artificial grains (not pores) are dropped from a random location onto the artificial sandbed which is assumed to be the border of our core sample). After those random procedures, a model like in Figure 2-4 is generated. Note that, the spheres in the figure are grains not pores.

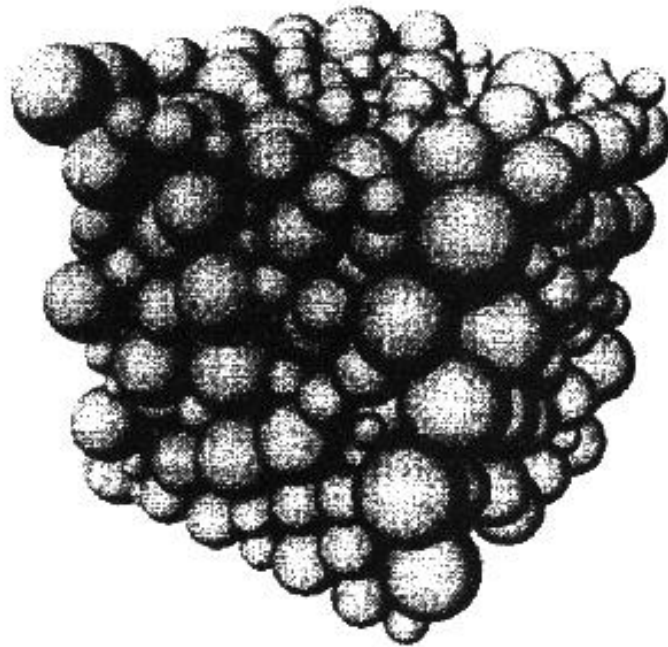


Figure 2-4: Sedimentation model for sandstone. (after Nilsen et. al., 1996)

The sedimentation process is followed by compaction. In compaction, all the z coordinates of grains are shifted vertically down by a factor called compaction factor. Figure 2-5 demonstrates this process.

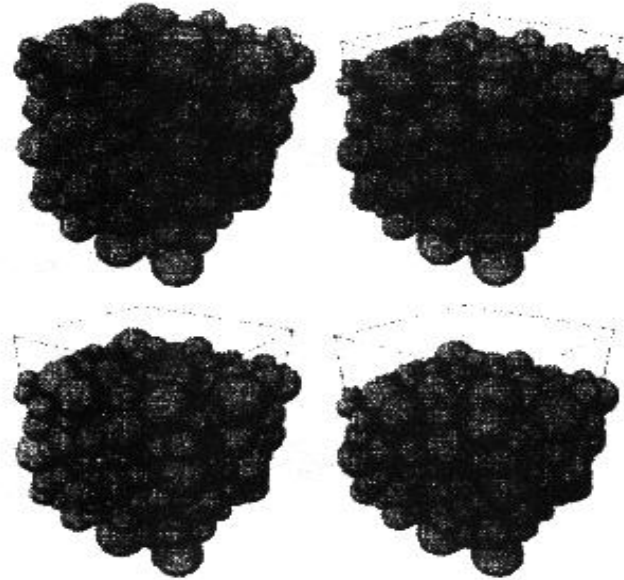


Figure 2-5: The simulated linear compaction process in sandstone for compaction factors of 0.0, 0.1, 0.2 and 0.3. (after Nilsen et. al., 1996)

After all these procedures, the model in hand can be considered as a correct model of grains. To convert this model to pore structure, first, the grain structure is converted to skeleton view (Figure 2-6) and then using this skeleton view, the location of pore bodies could be determined (At intersection points in the skeleton). The orientation of pore throats are also determined using this view.

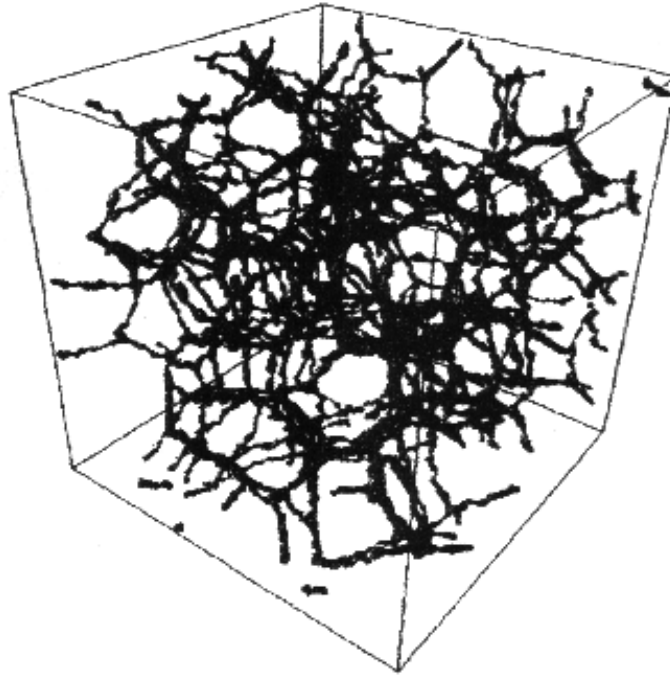


Figure 2-6: The skeleton view for a generated sandstone. (after Nilsen et. al., 1996)

After determining the pore body locations and the pore throat orientations, the generation of the pore network model is easy. Note that, although the reverse is possible, this approach calculates same radius values for a specific throat at the entrance of it to different pore bodies. Yet, the model is complex enough to accompany this drawback and give realistic results. The final model looks like the next figure (Figure 2-7).

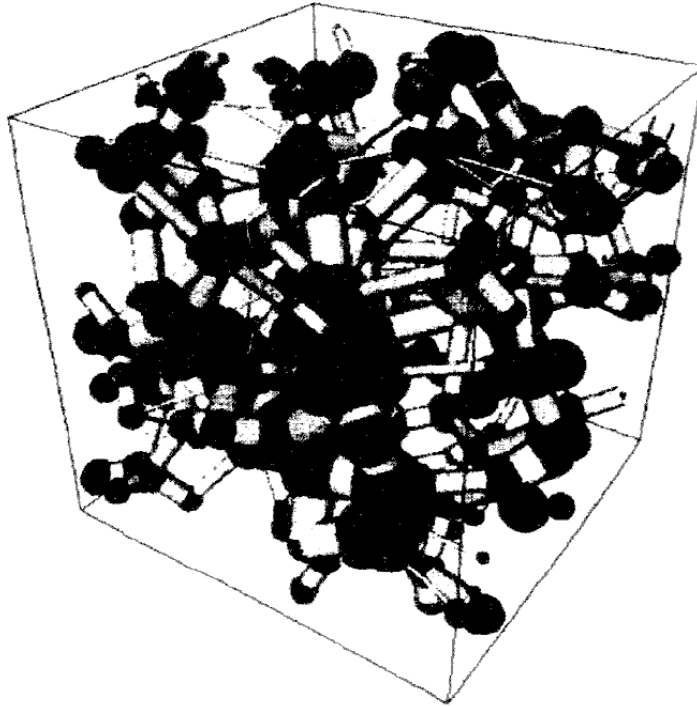


Figure 2-7: The ball (pore-body) and stick (pore-throat) representation of a 3-D network model. (after Nilsen et. al., 1996)

2.1.4. Basics of Network Models

Network models are often used to provide improved understanding of multiphase systems. Through such understanding, improved predictive capabilities may be achieved. In the case of network models, the predictions are usually parametric relationships that are required in continuum-scale descriptions of multiphase flow. These include relationships between capillary pressure and saturation and between relative permeability and saturation.

Two general types of network models may be identified: quasi-static displacement (Invasion Percolation) models and dynamic displacement models (Kamath et. al., 1996; Xu et. al., 1997). Quasi-static models impose a capillary

pressure on the network and calculate the final, static position of all fluid-fluid interfaces, neglecting dynamic aspects of pressure propagation within the model and interface dynamics. Dynamic models typically impose a specified inflow rate for one of the fluids and calculate the subsequent transient pressure response and associated interface positions. Because the practical application of network models often is the prediction of continuum-scale constitutive relationships, pore-scale network models are often constructed to mimic laboratory experiments used to measure constitutive relationships. In hydrology, a common laboratory experiment is the quasi-static determination of the relationship between capillary pressure and saturation. For a two-fluid system, this involves controlling the pressure of each fluid phase and measuring the fluid saturation as a function of pressure differences. A network model simulation of such an experiment begins with specification of the lattice properties, including throat-size and body-size probability distributions and coordination number. For primary drainage, all sites and bonds are initially filled with wetting fluid. One side of the lattice is assumed to be in contact with a reservoir of wetting fluid at pressure P_w , the other side is placed in contact with a reservoir of nonwetting fluid at pressure P_{nw} . Initially, $P_w = P_{nw}$. Capillary pressure is then increased incrementally by decreasing P_w or increasing P_{nw} , and the Laplace equation, modified to account for pore geometry is applied to each interface to test whether or not the interface is stable. If it is unstable (that is, the pore throat in which the interface resides is too large to support the imposed capillary pressure), the interface is advanced through the pore throat and the adjoining pore body (because pore bodies are required to be larger than adjoining pore throats). The interface eventually reaches a stable position at a pore throat with sufficiently small radius to support the imposed capillary pressure. This procedure is repeated for a series of step-wise changes in capillary pressure. At the end, some wetting fluid may remain in regions that became surrounded by invading fluid during the displacement process and is

thus isolated, i.e. ‘trapped’ from its reservoir, if such a trapping mechanism is allowed in the algorithm. Otherwise, wetting fluid remains behind only as pendular rings, although the geometry of network models often does not account for such pendular fluid. Because the pore throats control drainage, this algorithm is analogous to a ‘bond percolation’ problem.

Imbibition is achieved by decreasing the capillary pressure and allowing the wetting fluid to re-enter the network from its reservoir. This process is controlled by the pore bodies, because wetting fluid invades the smaller pore throats more easily than it does the larger pore bodies. Pore throats also influence imbibition through the mechanism of snap-off, whereby wetting fluid in films and wedges may coalesce to fill the narrow throats. Because pore-body radii influence imbibition while pore throats control drainage, hysteresis between drainage and imbibition follows naturally from the size differences between pore bodies and pore throats. Scanning curves also arise naturally via reversal of the drainage or imbibition process prior to reaching residual fluid saturations. Interfacial areas may also be calculated during both drainage and imbibition.

Given fluid distributions at each capillary pressure, resistance to flow may be calculated for each phase by assigning hydraulic resistances to each pore throat, based on an assumption of laminar flow in a tube, or Poiseuille flow. An equivalent network resistance may then be calculated by analogy to electrical resistors. This provides an estimate of relative permeability. In this calculation, a fluid pressure is calculated at each pore body, and a volumetric flux is calculated in each pore throat. Given the velocity field, miscible transport properties may be calculated.

In dynamic network models, a given volume of invading fluid is injected over a time step. Poiseuille flow is typically assumed in the pore throats, sometimes in conjunction with some assumption regarding motion of the ‘contact line’ that forms along the intersection points between the two fluid phases and the solid. At each time step, pressures are calculated at each pore body, and displacement decisions (interface propagation) are based on pressure difference rules that include capillary pressure.

2.1.5. Running the Pore Network Model

There are several different run procedures in the literature for 2D and 3D pore network models. Unfortunately, none proves to be superior. Yet, the 2D algorithm of Koplik and Lasseter (1985) seems to be one of the most advanced algorithms for 2D networking. Their paper is referenced by several authors; and, the results they had acquired are satisfactory (Karaman et. al., 1998). For a dominating 3D algorithm, there seems to be only one possibility: The algorithm presented in Lee et. al (1995) or very similar algorithms with some differences like throat distribution, etc. are used widely in literature (Nilsen et. al., 1996; Kantzas et. al. 1988; Kharabaf and Yortsos, 1996; Billiotte et. al., 1990; Satik and Yortsos, 1995). A brief summary of Lee and his friends’ algorithm is given below. That particular algorithm is chosen on purpose as,

- It is referenced by almost all of the papers written in 1996, and 1997; and, listed in the references part of this report.
- It is the most notable application of pore network models using high number of pores (about half million nodes); and, running on a super parallel computer.

- The algorithm is explained in details for imbibition process (Which is something very rare in the rest of the literature; generally, drainage is chosen to be examined (Nilsen et. al., 1996; Kantzas et. al. 1988; Kharabaf and Yortsos, 1996; Billiotte et. al., 1990; Satik and Yortsos, 1995)).

2.1.5.1. Capillary and Gravitational Forces

The pressure difference between bulbs and capillary forces in throats and bulbs will determine the flow direction. For instance, two pores *i* and *j* in Figure 2-8 respectively contain wetting and non-wetting phases and the circular tubes radius between the two pores is r_{ij} .

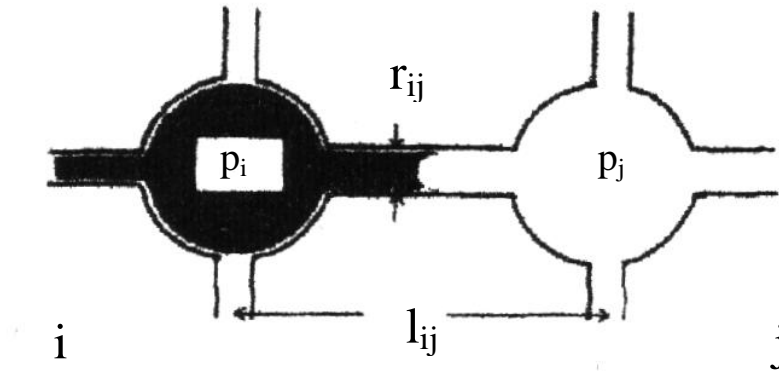


Figure 2-8: Imbibition in pores. (after Lee et. al., 1995)

When the surface tension between two fluids is σ and the contact angle is θ , the capillary pressure ($P_{c,ij}$) in the tube having the length of l_{ij} becomes,

$$P_{c,ij} = \frac{2\sigma \cos\theta}{r_{ij}} \quad (1)$$

Inside a pore, the wetting and non-wetting fluids can have different pressure due to the capillary force of the pore,

$$P_{p,j} = \frac{2\sigma \cos\theta}{r_{p,j}} \quad (2)$$

The capillary forces in the pores are generated by the random variable $r_{p,i}$ and the Young's formula.

In the imbibition process, the throat is originally occupied by the non-wetting fluid. When there is a gravitational force due to fluid density differences, the hydrodynamic potential difference can differ depending on what fluid occupies the throat. Recall that, although the fluid volume is contained in the bulb, the gravitational potential difference is proportional to the distance between two bulbs, which is equivalent to the throat length, l_{ij} . Hence, the potential difference should overcome the maximum barrier depending on the gravity direction and the flow direction. For the imbibition process as shown in Figure 2-8, the criterion for the wetting fluid to flow from pore i to pore j is,

$$P_i - P_{p,i} - P_j + \rho g \cdot l_{ij} > -P_{c,ij} \quad (3)$$

$$P_i - P_{p,i} - P_j + P_{p,j} + \rho_w g \cdot l_{ij} > 0 \quad (4)$$

The Eq. 3 indicates the potential criterion that the wetting phase can flow through the tube. The fluid density can be either that of the wetting or non-wetting phase fluid depending on which density gives a more strict criterion in Eq. 3. Eq. 4 presents the conditions for the flow of the wetting phase between two pores. Note that, the pore pressure is defined as a non-wetting phase pressure. The fluid in the capillary is initially the invaded fluid (i.e., the non-

wetting fluid for the imbibition process) and the fluid is replaced by the invading fluid. Therefore, the pressure difference should be big enough to overcome these two inequality criteria.

When a pore contains a mixed fluid, the simple ‘affinity rules’ were employed to determine which phase will flow between two pores: If the neighboring pore contains both wetting and non-wetting fluids, the wetting fluid will flow; and, if the neighboring cell has a single phase fluid, that fluid will flow between two pores. These rules can be expressed as follows:

1. If $S_{wi} > 0.001$ and $S_{wj} > 0.001$, the wetting phase will flow.
2. If $S_{wi} < 0.999$ and $S_{wj} < 0.001$, the non-wetting phase will flow

Note that, the saturation less than 0.001 or larger than 0.999 is considered as single phase. Without this approximation, many small time steps will be needed to imbibe pores with $S_w > 0.999$.

2.1.5.2. Conductivity

As the volume in the throat is neglected in this model, the transient period, in which the interface moves inside the throat, is not included in the conductivity calculations. Assuming the flow in the throat is Poiseuille, the flow rate of phase α between two pores is determined by,

$$Q_{\alpha,ij} = T_{\alpha,ij} (P_{\alpha,i} - P_{\alpha,j}) \quad (5)$$

where

$$P_{o,i} = P_i \quad P_{w,i} = P_i - P_{p,i} \quad (6)$$

$$T_{\alpha,ij} = \frac{\pi r_{ij}^4}{8\mu_{\alpha} l_{ij}} \quad (7)$$

Note that the criteria in the previous section (Capillary and Gravitational Forces) will allow no flow of a single flow in the tube.

2.1.5.3. Material Balance

Assuming incompressible flow, the material balance for each node (i.e., the node (i,j,k) in a Cartesian network) can be expressed by,

$$0 = T_{x,i,j,k}^{\alpha} (P_{i+1,j,k}^{\alpha} - P_{i,j,k}^{\alpha}) + T_{x,i-1,j,k}^{\alpha} (P_{i-1,j,k}^{\alpha} - P_{i,j,k}^{\alpha}) + T_{y,i,j,k}^{\alpha} (P_{i,j+1,k}^{\alpha} - P_{i,j,k}^{\alpha}) + T_{y,i,j,k}^{\alpha} (P_{i,j-1,k}^{\alpha} - P_{i,j,k}^{\alpha}) + T_{z,i,j,k}^{\alpha} (P_{i,j,k+1}^{\alpha} - P_{i,j,k}^{\alpha}) + T_{z,i,j,k}^{\alpha} (P_{i,j,k-1}^{\alpha} - P_{i,j,k}^{\alpha}) \quad (8)$$

The $T_{x,i,j,k}^{\alpha}$, $T_{y,i,j,k}^{\alpha}$, and $T_{z,i,j,k}^{\alpha}$ are the α phase conductivities in x, y and z directions, respectively.

2.1.5.4. Implementation for Solving

The linear pressure equation is solved by a point relaxation method with line correction (Watts 1971, 1973). The point relaxation method can be derived from Eq. 8.

$$\text{SUM T} = T_{x,i,j,k} + T_{x,i-1,j,k} + T_{x,i,j,k} + T_{x,i,j-1,k} + T_{x,i,j,k} + T_{x,i,j,k-1} \quad (9)$$

$$\begin{aligned} \text{SUM TP} = & T_{x,i,j,k} \cdot P_{i+1,j,k} + T_{x,i-1,j,k} \cdot P_{i-1,j,k} + T_{x,i,j,k} \cdot P_{i,j+1,k} + T_{x,i,j-1,k} \cdot P_{i,j-1,k} \\ & + T_{x,i,j,k} \cdot P_{i,j,k+1} + T_{x,i,j,k-1} \cdot P_{i,j,k-1} \end{aligned} \quad (10)$$

$$P_{i,j,k} = \text{SUMTP} / \text{SUMT} \quad (11)$$

It is well known that the point relaxation method is too slow or ineffective in reducing low frequency errors. This can be remedied by following the relaxation with an averaging scheme that is a backward linear correction for each cross-section, leading to a one-dimensional or line equation that can be solved serially backwards from the outlet to the inlet.

$$\Delta P_k = \frac{1}{\sum_{i,j} T_{x,i,j,k}} \left(\sum_{i,j} T_{x,i,j,k} (P_{i,j,k} - P_{i,j,k+1}) - Q \right) \quad (12)$$

$$P_{i,j,k+1} = P_{i,j,k} + \Delta P_k \quad (13)$$

Note that this line correction ensures the material balance between cross-sections along the flow-direction. With line correction it converges within 20-30 iterations, whereas without it, it fails to converge.

As the displacing fluid moves in from the cross-section of $k=0$, the pressure field will vary very little away from the displacing front. The point relaxation computations, which are local, can exploit this fact and can be confined to the invaded domain, thereby substantially reducing the computational time. The line correction is added every 3-5 point relaxation steps.

2.2. Artificial Neural Networks

Artificial Neural Networks (ANNs) are one of the most popular methods of artificial intelligence and are inspired by the adaptive and massively distributed structure of biological neural networks similar to the ones found in the human brain.

2.2.1. What is a Neural Network?

Where does intelligence emerge? There are two important ways to answer this question from the computational point of view. One is based on symbolism, and the other, based on connectionism. The former approach models intelligence using symbols, while the latter using connections and associated weights. Evolving by different routes, they both have achieved many successes in practical applications. (Fu, 1994)

Work on neural networks, has been motivated right from its inception by the recognition that the brain computes in an entirely different way from the conventional digital computer (Haykin, 1994). Anyone can see that the human brain is superior to a digital computer at many tasks. A good example is the processing of visual information: a one-year-old baby is much better and faster at recognizing objects, faces and so on than even the most advanced artificial intelligent system running on the fastest supercomputer. The brain has many other features that would be desirable in the artificial systems. It is robust, fault tolerant, flexible, highly parallel, small, compact and it can easily adjust to a new environment by learning and deal with information that is fuzzy, probabilistic, noisy or inconsistent. In addition, it does not have to be programmed in any programming language such as Pascal, Fortran or C; and it also needs very little power contrary to the digital computers do (Hertz et. al., 1991).

2.2.2. Inspiration from the Neuroscience

The struggle to understand the brain owes much to the pioneering work of Ramón y Cajál (1911), who introduced the idea of neurons as structural constituents of the brain. Today we know that a human brain contains some 100 billion (10^{11}) neurons with about 1,000 to 10,000 connections each (resulting in

a total of 10^{14} - 10^{15} interconnections). Although one neuron is about 106 times slower than a transistor of a computer (neuron: 10^{-3} sec, transistor: 10^{-9} sec) the massively parallel processing capability of the brain gives it a much higher efficiency. For example, it takes about 100-200 msec for the brain to recognize a familiar face on a photograph; this simple task already causes great problems to the computer - if it can do it at all. (Figure 2-9)

Each neuron receives its input from neighboring neurons over special nerve fibers, called dendrites. These input signals can be either excitatory or inhibitory. Their sum builds up an electrochemical potential inside the cell body of the neuron. When this potential exceeds a certain threshold the neuron fires and sends signals along the output fiber, the so-called axon. The axon – each cell only has one – normally is quite short, but can reach several meters in length. It ends in synapses. These are the connections to the dendrites of another neuron or directly to another neuron. The synapses are thought to be mainly responsible for learning.

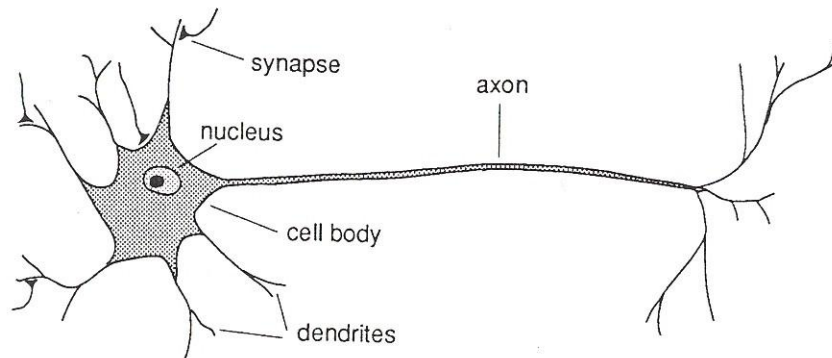


Figure 2-9: Schematic drawing of a typical neuron. (after Hertz et. al., 1991)

2.2.3. Neural Computational Models

The neural network contains a large number of simple neuron-like processing elements and a large number of weighted connections between the elements. The weights on the connections encode the knowledge of a network. Though biologically inspired, many of the neural network models developed do not duplicate the operation of the human brain. Some computational principles in these models are not even explicable from biological viewpoints (Fu, 1994).

For example, the processes that recognizing human faces and understanding the speech, the current artificial intelligence systems can not do better than the human brain even they can do higher order more complicated arithmetic calculations so rapidly. It can be said, for the cause of this situation, that the structure of brain is somehow suited to above-mentioned processes and not suited to tasks such as high-speed arithmetic calculations.

Although the brain cell responds much more slowly than a digital one, as operating in parallel, the slowly responsive cell bodies in themselves can reach a solution quickly.

Another feature exhibited by the brain is the associative type of memory. It naturally associates one thing with another therefore it can access information based on contents rather than on sequential addresses as in the digital computer. The associative memory accounts for fast information retrieval and permits partial or approximate matching (Fu, 1994).

With the distinct performance features, many neural network models were built such as Hopfield net, Boltzmann machine, Backpropagation net, etc.

All of the mentioned and the others have discrete features and were developed by using different models.

In “Classification Models”, neural network classifies a given object presented to it according to the output activation. The perceptrons are classified as their holding number of hidden layers such as Single Layer or Multilayer.

The “Association Models”, such as Hopfield Nets and Bidirectional Associative Memories, in which a memory can be completed or corrected upon retrieval by-self association given a partial or corrupted input, or relating the input vector to another different one and generalization over similar inputs.

The “Optimization Model” offers solutions to various combinatorial optimization problems, which often lack efficient solutions on a digital computer (Hopfield Nets, Boltzman Machines etc.).

The “Self-organization” model (for example Kohonen Network) is effective for dealing with problems whose algorithms are too complicated to define, for example, modeling of irregular surfaces. The term self-organization refers to the ability to learn and organize information without being given correct answers for input patterns. Thus, self-organizing networks perform unsupervised learning. This computational model may serve to explain some neurobiological phenomena such as how a baby learns since it does not know what is correct (Fu, 1994).

2.2.4. Learning Process

There are too many notions associated with “learning” to justify defining the term in a precise manner. Moreover, the process of learning is a matter of viewpoint, which makes it all the more difficult to agree on a precise

definition of the term. For example, learning viewed by a psychologist is quite different from learning in a classroom sense (Haykin, 1994).

In the literature lots of definitions were given to learning. Two of them are followed:

“Learning is a process by which the free parameters of a neural network are adapted through a continuing process of simulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place.” (Haykin, 1994)

And the definition suggested by Herbert Simon, “Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time” (Fu, 1994)

In the domain of artificial neural networks, learning is defined to be any change in the memory and it can be divided into two categories as unsupervised and supervised learning.

2.2.4.1. Unsupervised (or Self-organization) Learning

It is the process that incorporates no external teacher and relies upon only local information and internal control. The neural networks which employ this type of learning self-organize the presented data and discover its emergent collective properties. Afterwards the networks of this type should discover internal categories, correlations or patterns for the input data in a more abstract and less defined manner.

2.2.4.2. Supervised (or Associative) Learning

In this learning method, the network is trained with a group of input and corresponding output data until the network output for an input set is as close as possible to the known correct output. These mentioned data (pair of input-output data) should be provided by an external source as opposed to the unsupervised learning type. Perceptron, ADALINE and the back-propagation are the examples of supervised learning.

Karaman and Demiral (2003a, August 2004) have used the back-propagation type of learning. They have adjusted the connection weights by concerning the minimum total error between the desired and computed values of each node in the outlet. Same philosophy has been used in this study.

2.3. NMR Relaxation Mechanisms

Petrophysical information (e.g., porosity, pore-size distribution, bound water and permeability) can be extracted from NMR relaxation measurements. For the proper application of NMR in formation evaluation it is critical to understand the nature of the NMR relaxation of fluids in pores.

2.3.1. NMR Relaxation Mechanisms of Fluids in Pores

T1 (Longitudinal) and T2 (Transverse) relaxations are caused by magnetic interactions between protons.

T1 occurs when a precessing proton system transfers energy to its surroundings. Donor proton relaxes to low-energy-state, in which they precess along the direction of B₀.

T2 contributed by the same energy transfer and also by dephasing without energy transfer to surrounding. Therefore: $T_2 \leq T_1$.

In general:

- For protons in solid: $T_2 \ll T_1$,
- For protons in reservoir fluid:
 - when the fluid is in a homogenous static magnetic field and a Carr-Purcell-Meiboom-Gill (CPMG - 90° pulse followed by a long series of 180° pulses (Kleinberg et. al., 1993)) measurement is used: $T_2 \approx T_1$
 - when it is in a gradient magnetic field: $T_2 < T_1$
- When a wetting fluid fills a porous medium, both T_2 and T_1 are decreased; the relaxation mechanisms are different from those of the protons in either the solid or the fluid.

For fluid in pores, three independent Relaxation Mechanisms are involved:

1. Bulk fluid processes that affects both T_1 and T_2 ,
2. Surface relaxation that affects both T_1 and T_2 ,
3. Diffusion in the presence of magnetic field gradient that only affects T_2 .

All three processes act in parallel; then T_1 and T_2 can be expressed by Eq. 14 and Eq. 15 respectively:

$$\frac{1}{T_1} = \frac{1}{T_{1bulk}} + \frac{1}{T_{1surface}} \quad (14)$$

T1: the measured longitudinal relaxation time of the pore fluid

T1bulk: T1 relaxation time of the pore fluid as it would be measured in a container

T1surface: T1 relaxation time of the pore fluid resulting from surface relaxation

$$\frac{1}{T_2} = \frac{1}{T_{2bulk}} + \frac{1}{T_{2surface}} + \frac{1}{T_{2diffusion}} \quad (15)$$

T2: transverse relaxation time of the pore fluid as measured by a CPMG sequence

T2bulk: T2 relaxation time of the pore fluid as it would be measured in a container

T2surface: T2 relaxation time of the pore fluid resulting from surface relaxation

T2diffusion: T2 relaxation time of the pore fluid as induced by diffusion in the magnetic field gradient

The relative importance of the three relaxation mechanisms depends on:

- The type of fluid in the pores (water, oil or gas),
- The size of the pores,
- The strength of the surface relaxation,

- The wettability of the rock surface.

In general:

- For brine; T2 is dominated by T2surface,
- For heavy oil; T2bulk is the main contributor of T2,
- For intermediate-viscosity and light oil; T2 is a combination of T2bulk & T2diffusion and is dependent on viscosity,
- For gas; T2 is dominated by T2diffusion.

2.3.1.1. Bulk Relaxation

Bulk relaxation is the intrinsic relaxation property of a fluid. It is controlled by the physical properties of the fluid (e.g. Viscosity and chemical composition) and can be measured by placing the fluid in a large container (thereby eliminating surface relaxation) and then subjecting the fluid to a homogeneous magnetic field and a CPMG pulse sequence.

It is also affected by other environmental conditions such as temperature and pressure.

Bulk Relaxation Times (in seconds) for water (Eq. 16), gas (Eq. 17) and oil (Eq 18):

$$T_{1bulk} \cong 3 \left(\frac{TK}{298\eta} \right) \quad \text{and} \quad T_{2bulk} \cong T_{1bulk} \quad (16)$$

$$T_{1bulk} \cong 2.5 \times 10^4 \left(\frac{\rho_g}{T_K^{1.17}} \right) \quad \text{and} \quad T_{2bulk} \cong T_{1bulk} \quad (17)$$

$$T_{1bulk} \cong 0.00713 \left(\frac{T_K}{\eta} \right) \quad \text{and} \quad T_{2bulk} \cong T_{1bulk} \quad (18)$$

$T_K = \text{temperature (K)}$
 $\eta = \text{fluid viscosity (cp)}$
 $\rho_g = \text{gas density (gm/cc)}$

2.3.1.2. Surface Relaxation

It occurs at the fluid-solid interface. Governing equations are as follows:

$$\frac{1}{T_{1surface}} = \rho_1 \left(\frac{S}{V} \right)_{pore} \quad (19)$$

$$\frac{1}{T_{2surface}} = \rho_2 \left(\frac{S}{V} \right)_{pore} \quad (20)$$

$\rho_2 = T_2$ surface relaxivity (T_2 relaxing strength of the grain surfaces)

$\rho_1 = T_1$ surface relaxivity (T_1 relaxing strength of the grain surfaces)

$(S/V)_{pore} = \text{ratio of pore surface to fluid volume}$

For simple shapes, S/V is a measure of pore size. For example, for a sphere, $S/V = 3/r$ where r is the radius of the sphere).

The dominant term in T1 and T2 surface relaxation times is surface relaxivity. It varies with mineralogy. For example, carbonate surfaces exhibit weaker surface relaxivity than quartz surfaces, and estimates can be determined in laboratory.

Fluids controlled by surface relaxation exhibit T2 times that are not dependent on temperature and pressure. For this reason, laboratory NMR measurements at room conditions are commonly used to calibrate the formulas that are used to estimate petrophysical properties such as permeability and bound water.

Models determined from laboratory measurements can be directly transferred to the interpretation of NMR logging data, because the measurement acquisition process used in the laboratory is identical to the one used during logging. Thus this transferability simplifies the interpretation process.

2.3.1.3. Diffusion-induced Relaxation

When gas, light oil, water and some medium-viscosity oils are in a gradient magnetic field and are subjected to a CPMG sequence with long inter-echo spacing, they exhibit significant diffusion-induced relaxation. For these fluids, T2diffusion relaxation time constant, associated with the diffusion mechanism, becomes an important tool for their detection.

When a significant gradient exists in the static magnetic field, molecular diffusion increases the T2 relaxation rate ($1/T2$). And, diffusion has no influence on the T1 relaxation rate ($1/T1$).

The diffusion-induced relaxation rate:

$$\frac{1}{T_{2diffusion}} = \frac{D \gamma^2 G^2 TE^2}{12} \quad (21)$$

D = molecular diffusion coefficient

γ = gyromagnetic ratio of a proton

G = field-strength gradient (G/cm)

TE = inter-echo spacing used in the CPMG sequence

Physical properties such as viscosity and molecular composition, environmental conditions, temperature and pressure affect the diffusion.

Gas, oil and water diffusion coefficients are as follows:

$$D_g = 8.5 \times 10^{-2} \left(\frac{TK}{\rho_g} \right) \times 10^{-5} \text{ cm}^2/s \quad (22)$$

$$D_o = 1.3 \left(\frac{TK}{298\eta} \right) \times 10^{-5} \text{ cm}^2/s \quad (23)$$

$$D_w = 1.2 \left(\frac{TK}{298\eta} \right) \times 10^{-5} \text{ cm}^2/s \quad (24)$$

$D_{g,o,w}$ increase with temperature (viscosity (η) of oil and water decreases with temperature); D_g decreases with increasing pressure because gas density (ρ_g) increases with pressure (P) and D_o varies significantly because

different oils exhibit a wide range of molecular composition that results in a wide range of viscosities.

Molecular movement in wetting fluids is often restricted:

- at the interface between rock grains and fluids,

and/or

- by the interfacial tension between fluids.

Then, the diffusion coefficient for a fluid in a rock differs from the diffusion coefficient of the bulk fluid at the same temperature and pressure. Except the HC gas, diffusion effects become unimportant at short TE (inter-echo spacing) for most fluids. If necessary, diffusion effects can be minimized by selecting the appropriate TE for the desired application.

Integration of the Eqs. 16 and 21 into Eqs. 14 and 15 yields:

$$\frac{1}{T1} = \frac{1}{T1_{bulk}} + \rho_1 \left(\frac{S}{V} \right)_{pore} \quad (25)$$

$$\frac{1}{T2} = \frac{1}{T2_{bulk}} + \rho_2 \left(\frac{S}{V} \right)_{pore} + \frac{D \left(\frac{S}{V} \right)_{pore}^2}{12} \quad (26)$$

2.3.2. Multi-exponential Decay

In general reservoir rocks exhibit:

- a distribution of pore sizes,

and

- frequently contain more than one fluid type.

Therefore, the spin echo train recorded with CPMG sequence does not decay with a single T2 value but instead with a distribution of T2 values:

$$M(t) = \sum_i M_i \exp\left(-\frac{t}{T_{2i}}\right) \quad (27)$$

$M(t)$ = measured magnetization at time t

$M_i(0)$ = initial magnetization from I^{th} component of relaxation

T_{2i} = decay constant of the I^{th} component of transverse relaxation

The summation is over the whole sample (i.e. All pores and all different fluid types)

Figure 2-10 illustrates the multi-exponential decay character of a porous medium containing pores of different size but similar shape and a single wetting fluid. When formation is only brine saturated and short inter-echo spacing is used, then T2 is directly proportional to pore size.

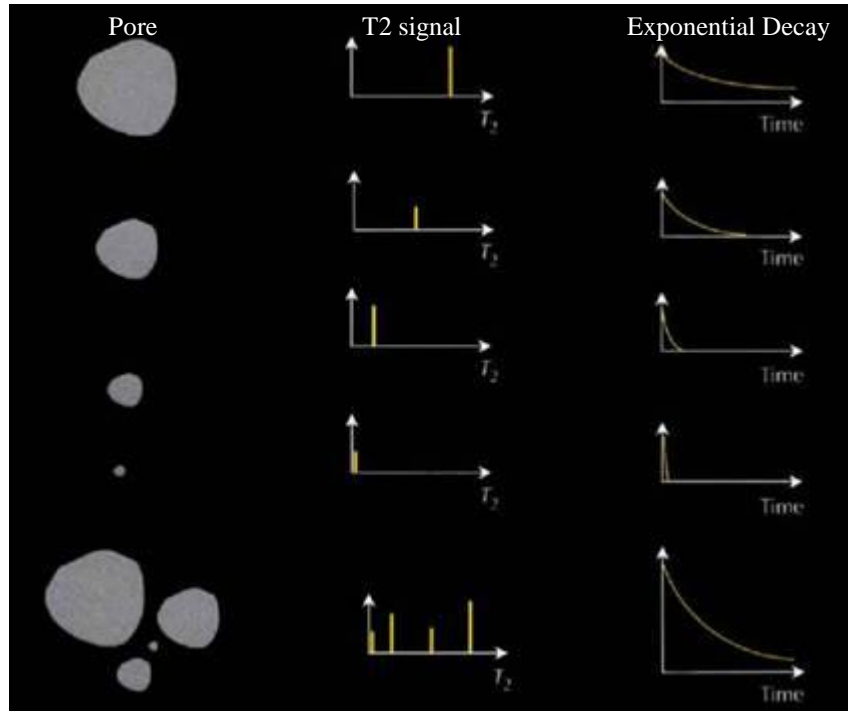


Figure 2-10: Illustration of multi-exponential decay character of a porous medium. (after Coates et. al., 1999)

As seen from Figure 2-10, the largest pores have the lowest S/V and thus the longest T2; smallest pores have the highest S/V and thus the shortest T2.

A 100% water saturated pore has a single T2 value that depends on pore size and its spin-echo train exhibits a single-exponential decay that also depends on the pore size.

Multiple pores at 100% water saturation have multiple T2 values that depend on pore size and their spin-echo train exhibits multi-exponential decay that also depends on the pore sizes.

If the pores are saturated by only water the corresponding magnetization equation (Eq. 27) becomes:

$$M = \sum M_i e^{-\rho_2 \left(\frac{S}{V} \right)_i t} \quad (28)$$

If the pores are partially saturated (i.e. The pores contain oil and/or gas in addition to water) the corresponding magnetization equation is as follows:

$$M = \sum M_i e^{-\rho_2 \left(\frac{S}{V} \right)_i t} + M_{oil} e^{-t/T_{2oil}} + M_{gas} e^{-t/T_{2gas}} \quad (29)$$

M_{oil} = magnetization produced by oil protons in the pores

M_{gas} = magnetization produced by gas protons in the pores

T_{2oil} = T_2 of oil measured with CPMG sequence

T_{2gas} = T_2 of gas measured with CPMG sequence

Eq. 29 assumes that the medium is water-wet rock, and spin-echo decays of both oil & gas can be characterized by using a single-exponential expression which represents the bulk and diffusion relaxation properties of non-wetting fluids.

Actually, many crude oils are composed of multiple HC types thus have complex decay spectra that must be represented by multi-exponential sums.

Furthermore, when oil or gas occupies a part of the pore in addition to water, water saturation decreases while S/V decreases and T_2 spectrum no longer represents a pore-size distribution because of the bulk response of the

non-wetting fluid. But, the porosity in these pores is accounted for in the non-wetting bulk fluid response; thus the porosity is not affected although the pore-size distribution is distorted.

2.3.3. Echo-fit for T2 distribution

Echo-fitting is one of the most important steps in NMR data processing. It is a mathematical inversion process and it uses a multi-exponential model assuming T2 distribution consists of discrete relaxation times corresponding to porosity components although T2 distribution is a continuous function in order to fit the echo train easily.

$$\begin{aligned}
 \text{echo}(1) &= \phi_1 e^{-\frac{1}{T_{2,1}}} + \phi_2 e^{-\frac{1}{T_{2,2}}} + \dots + \phi_m e^{-\frac{1}{T_{2,m}}} + \text{noise} \\
 \text{echo}(2) &= \phi_1 e^{-\frac{2}{T_{2,1}}} + \phi_2 e^{-\frac{2}{T_{2,2}}} + \dots + \phi_m e^{-\frac{2}{T_{2,m}}} + \text{noise} \\
 &\vdots \\
 \text{echo}(n) &= \phi_1 e^{-\frac{n}{T_{2,1}}} + \phi_2 e^{-\frac{n}{T_{2,2}}} + \dots + \phi_m e^{-\frac{n}{T_{2,m}}} + \text{noise}
 \end{aligned} \tag{30}$$

In this process the input is echo-train & the output is T2 distribution.

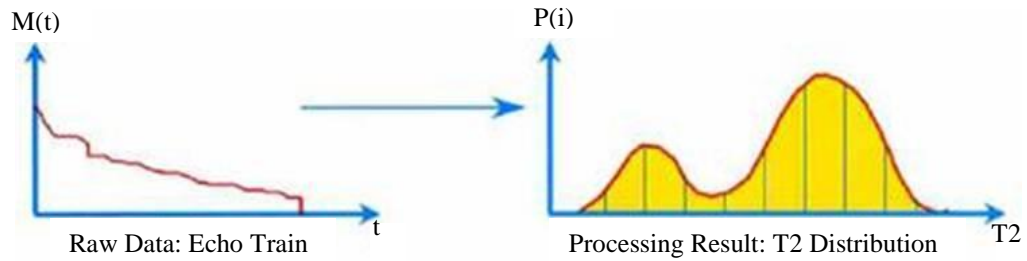


Figure 2.11: Echo-fit for T2 Distribution. (after Coates et. al., 1999)

2.3.4. Determination of Bulk Volume of Irreducible Water (BVI)

Currently, two methods are available for BVI determination:

1. **Cut-off BVI:** It is based on a fixed T2 value ($T_{2\text{cutoff}}$) that divides the T2 distribution into two components:
 - one consisting of pore sizes containing bound water,
 - the other consisting of pore sizes containing free fluids.

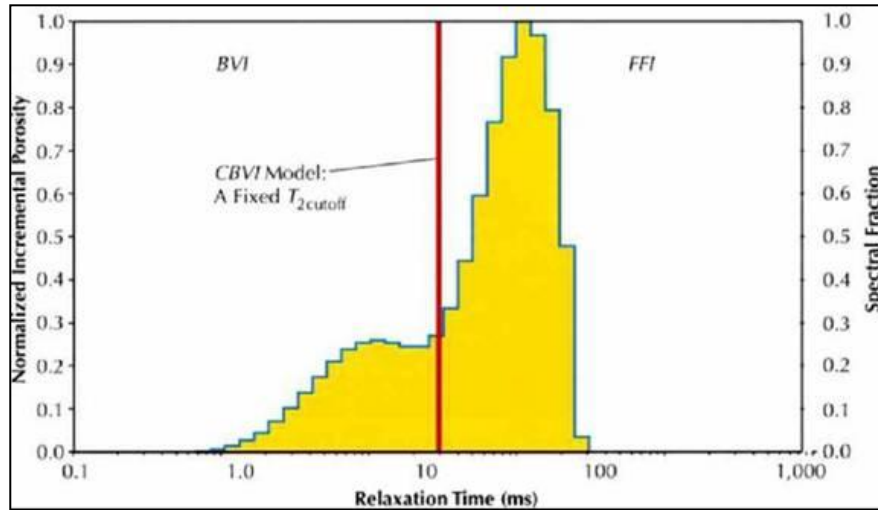


Figure 2.12: Cut-off BVI Model. (after Coates et. al., 1999)

2. **Spectral BVI:** It is based on the recognition that a given pore can contain both free and bound fluids.

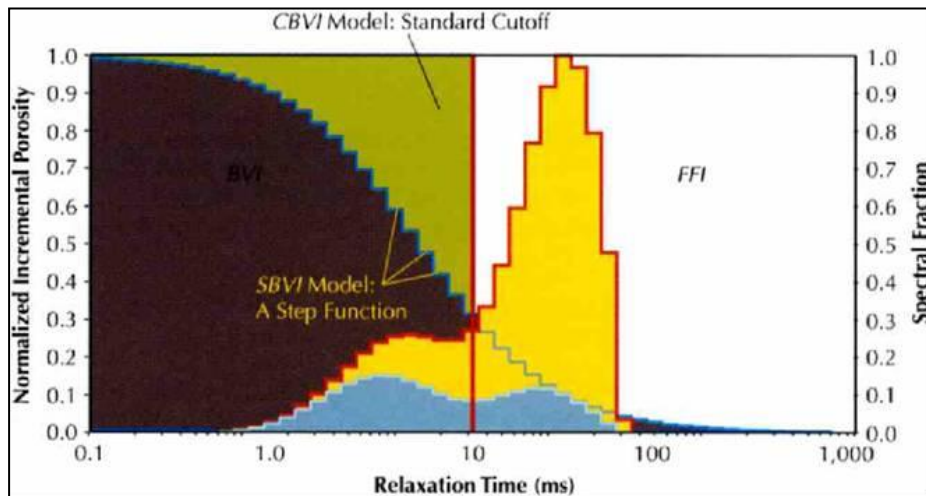


Figure 2.13: Spectral BVI Model. (after Coates et. al., 1999)

$$\begin{aligned}
S_{wirr,1}\phi_1 &= W_1\phi_{1,1} + W_2\phi_{1,2} + \dots + W_n\phi_{1,n} \\
S_{wirr,2}\phi_2 &= W_1\phi_{2,1} + W_2\phi_{2,2} + \dots + W_n\phi_{2,n} \\
&\vdots \\
S_{wirr,m}\phi_m &= W_1\phi_{s,1} + W_2\phi_{s,2} + \dots + W_n\phi_{m,n}
\end{aligned} \tag{31}$$

W_i are the weighting functions. $S_{wirr,i}$ are the water saturations of the samples at a chosen capillary pressure and ϕ_i are the porosities of the samples. In almost any pore model the percentage of bound water in a pore decreases as pore size increases then weighting function could be expressed as:

$$\frac{1}{W_i} \sim T_{2,i} \tag{32}$$

2.3.5. Magnetic Resonance Imaging Log (MRIL) Permeability Model

2.3.5.1. The Free-Fluid (or Coates) model

$$k = \left[\left(\frac{\phi}{C} \right)^2 \left(\frac{FFI}{BVI} \right) \right]^2 \tag{33}$$

C is dependent on the processes that created the formation and can be different for each formation. Free Fluid volume (FFI) & BVI from both CBVI or SBVI.

2.3.5.2. The Mean T2 (or SDR) Model

It works very well in zones containing only water and fails for hydrocarbon-bearing formations.

$$k = a T_{2gm}^2 \phi^4 \quad (34)$$

a is dependent on the formation and T_{2gm} is geometric mean of the T_2 distribution.

2.3.6. MRIL Porosity Model

The initial amplitude of the spin echo train (or area under the T_2 distribution curve) is proportional to number of H protons contained in the pore fluids. Thus, this amplitude can be calibrated to give a porosity value.

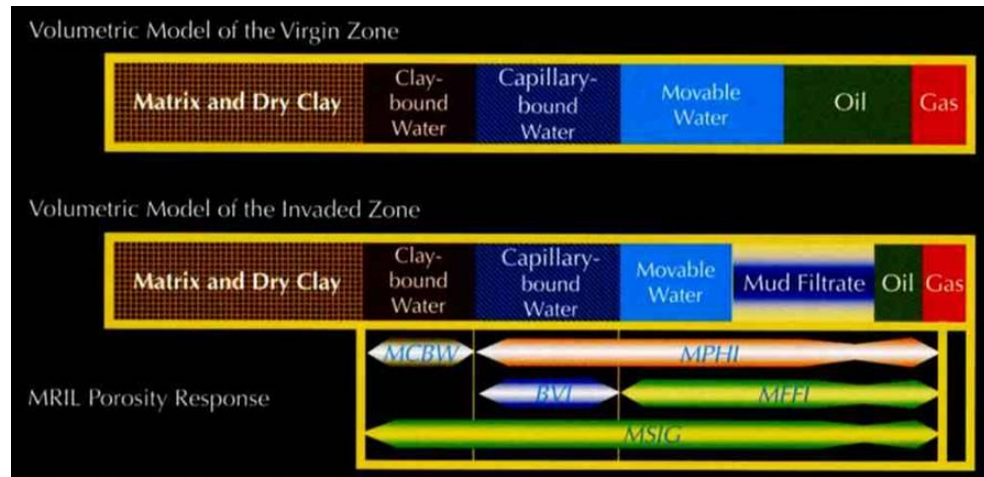


Figure 2.14: NMR porosity model for a water-wet formation. (after Coates et. al., 1999)

Sharp divisions between the porosity elements associated with free fluid index (MFFI), bulk volume irreducible (BVI) and clay-bound water (MCBW). It necessarily corresponds to sharp divisions in T_2 spectra.

In the case of clays, water associated with clays has a range of decay times that can overlap the ones for capillary-bound water; thus, a sharp division may not exist between MCBW and BVI in T_2 spectra.

Measured echo-train amplitude and thus effective porosity (MPHI) are affected by the hydrogen index, HI (i.e. measure of the density of Hydrogen atoms in the fluid). MPHI is calibrated to the protons in water having HI=1; if HI is significantly different from 1 then a correction to MPHI is required.

Both MPHI & MCBW; affected by TW (polarization time, i.e. the time between the stopping of the measurement of one echo train and the beginning of measurement of the next echo train). If it is too short, then effective porosity may be underestimated. So that, to achieve the complete polarization of the Hydrogen nuclei in the fluids it should be sufficiently long.

Both MPHI & MCBW; affected by TE (i.e. the time between the individual echoes in an echo train). As TE decreases, faster T2 relaxation components can be detected. Conversely, increasing TE causes the loss of fast T2 components, which in turn causes MPHI & MCBW to be too low.

CHAPTER 3

STATEMENT OF THE PROBLEM

In this thesis, it is aimed to develop a neural network model that predicts the two phase oil – water relative permeability. In order to achieve this a pore network model was developed. This pore network model takes pore size distribution, which was found by thin section imaging, as input for the same porous medium and calculates the subjected two phase oil – water relative permeability curves for primary drainage and imbibition displacement processes separately and T2 relaxation times for a porous medium.

Then, a neural network model was trained by using the relative permeability and T2 relaxation time data obtained from the developed pore network model. At the end, a robust neural network tool, which estimates the two phase oil – water relative permeability values for any given fluid configuration and also any given porous medium whose NMR data was known before, has been developed.

By having this neural network tool, the hypothetical aim has been accomplished and as a further step, it could be patched to NMR logging tool in order to omit the time consuming relative permeability experiments. By

this way, it is obvious that through out the logging measurement inside the well there will be the chance to have relative permeability data in real time.

CHAPTER 4

METHODOLOGY

As being inspired what has been done to reach porosity and permeability values of the reservoir rock – fluid systems by using the nuclear magnetic resonance imaging logging tools, it has been decided to calculate two phase oil – water relative permeability values. To do this, experimental and computer studies had to be done.

The final destination was forming a robust artificial neural network program (flow chart of the methodology has been given in the Figure 4.1) that inputs the synthetic data calculated from constructed pore network model. While generating the pore network model, real experimental data has to be taken into account to test the correctness of the model. In order to achieve a correct model that could represent the structure of the real core plugs and that could mimic the two phase oil – water displacements, it should be optimized by comparing the data obtained from itself with the experimental data obtained by using the same core plugs. While the implementation of this optimization process, pore sizes and also contact angles have been adjusted.

The core plugs have been taken from Research Center at Turkish Petroleum Corporation (TPAO). Basic and special core analyses have been

performed at this center on the same core plugs. This data sets have been used while pore network optimization.

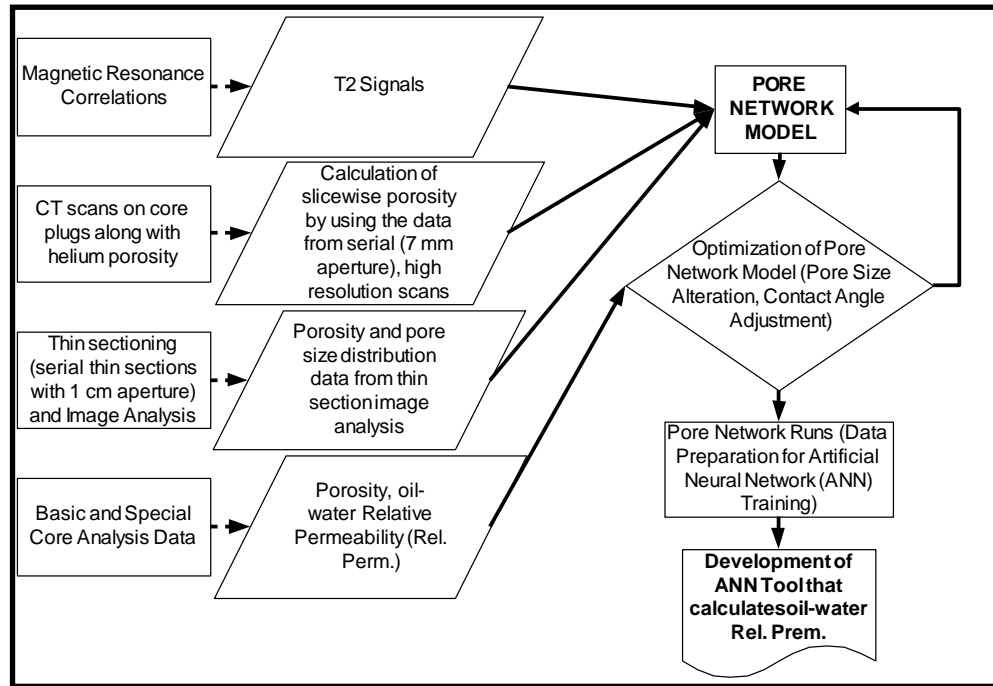


Figure 4.1: Flow chart of the methodology

Data to be used to construct the pore network model have been obtained by performing the CT scanning along with the helium porosimetry measurements and also thin section imaging. Serial thin sections with 1 cm aperture have been prepared by TPAO Research Center. After imaging study of the thin sections by using the microscope with 10x magnification, analysis on them has been carried out by using the ImageJ image analysis program.

A statistical investigation on the radius data obtained from the image analysis has been performed and then a pore network model has been constructed by taking into account the analysis results of thin section imaging. After reaching plausible results from the constructed model, several data generation runs have been implemented after putting a patch into the model that calculates the T2 transverse relaxation times. Those

calculations have been done by using the correlations available in the literature. These generated data were used to train constructed artificial neural network model. Afterwards, the model has been tested by using the available data separated from the dedicated training data set.

Step by step workflow scheme is detailed and the results are discussed in the following chapters.

CHAPTER 5

IMPLEMENTATION TO THE PORE NETWORK MODELLING

In this part of the study, in order to find the pore size and porosity distributions along the core plugs to be used to develop pore network model, Computerized Tomography (CT) imaging and thin section imaging have been performed. Then, a subroutine has been added to model to calculate the Nuclear Magnetic Resonance (NMR) response of the system.

5.1. CT and Thin Section Image Processing

The core plugs used in the experiments are numbered as 3-3, 4-1, 92-2 and 100-4. The rock types, reservoir and well numbers of them are given in Table 5-1.

Table 5-1: Core Plug Information

<i>Plug#</i>	<i>Reservoir</i>	<i>Well #</i>	<i>Rock Type</i>
3-3	Kuzey Marmara	1	Limestone
4-1	Berea	N/A	Sandstone
92-2	Yenikoy	32	Dolostone
100-4	Yenikoy	29	Dolomitic Limestone

In order to explain what have been done throughout the development phase of the pore network model, work flow on core plug 3-3 has been illustrated in this chapter. All of the results on the other core plugs are also given in Appendix D and Appendix E.

CT images of 6 different core plugs (Figure 5-1 and Figure D-1... Figure D-5) have been taken by using Philips Tomoscan 60/TX third generation scanner. Length between adjacent slices has been set to 7 mm and in order to calculate porosity values of core plugs, serial scans have been carried out with two different energy levels (i.e. 100 KeV and 130 KeV) at 250 mA tube current.

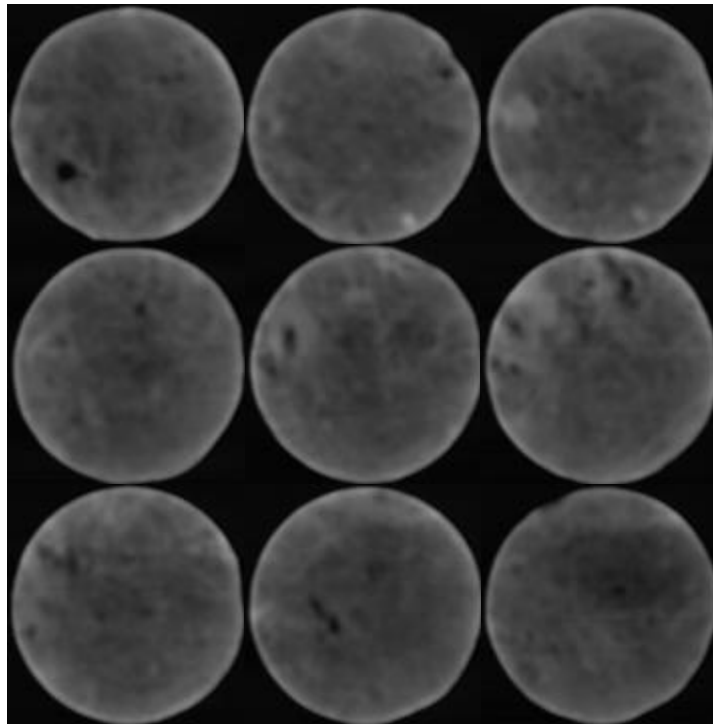


Figure 5-1: CT images of Core Plug 3-3 (Limestone)

In order to find porosity distribution along the core plugs, a quantitative determination of the porosity from measured attenuation coefficients proposed by Akin et al. (1996) has been used. Porosity values

obtained from helium porosity measurements are used in the proposed formulation (Eq. 5-1) and the calibration matrix CT number (N_M^i) is obtained by the back substitution.

$$\phi = 1 - \left(\frac{N_B^2 N_{air}^1 - N_B^1 N_{air}^2}{N_M^2 N_{air}^1 - N_M^1 N_{air}^2} \right) \quad (5-1)$$

The superscripts 1 and 2 represent the energy levels of 100 KeV and 130 KeV respectively. Bulk CT numbers (N_B^i) are obtained by CT measurements on core plugs at each mentioned energy levels; and air CT numbers (-1109.38 at 100 KeV and -1084.2 at 130 KeV) are calculated by scanning the core holder while empty.

Then, slice-wise porosity data plotted with the results obtained by helium porosimetry as shown in Figure 5-2 (The rest has been given in Appendix D – Figure D-6...Figure D-10).

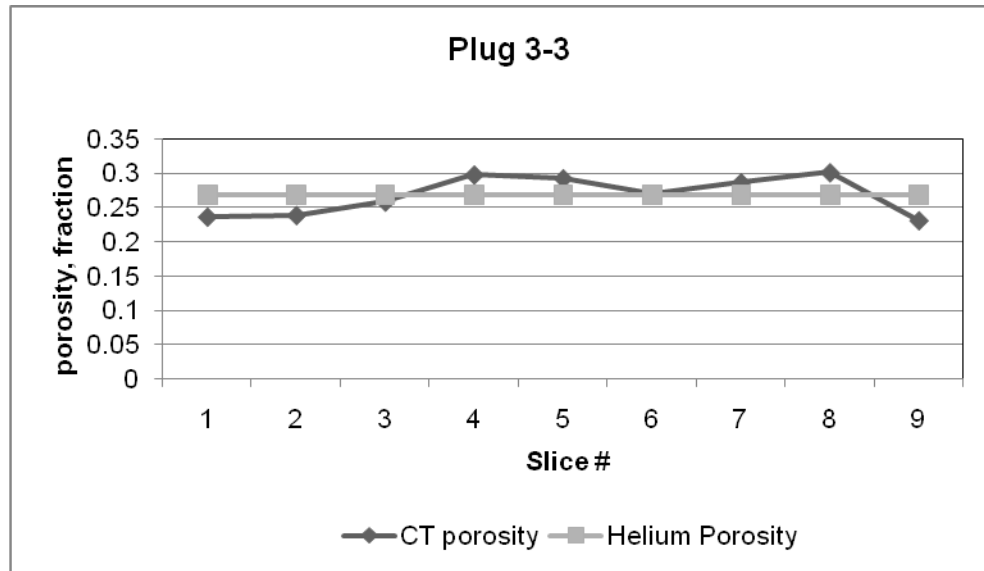


Figure 5-2: Porosity Distribution in Core Plug 3-3

The aim of CT measurement implementation on core plugs was to reach a bulk distribution of porosity inside the core plugs. By this way, basic

properties of pore network construction have been obtained. And pore network have been formed by taking into account this obtained porosity distribution with randomly distributed pore-body and pore-throat radii.

In order to have more robust pore network model, the primitive one developed after CT measurements phase should be revised by means of well organized pore-body and pore-throat radii and their distributions. In this manner, it was decided to cut thin sections from the core plugs and do image analysis on them to obtain the necessary data.

In thin sectioning part of the study, seven serial thin sections have been made on four different core plugs (plug 4-1, 92-2, 3-3 and 100-4) by Research Center at Turkish Petroleum Corporation (TPAO); and nine (3x3) images have been taken (with 10x magnification) on each thin section (Figure E-1...Figure E-3) by using the Leica DFC 280 microscope available in the Petroleum and Natural Gas Engineering Image Processing Laboratory at Middle East Technical University. The images taken from the first thin section of core plug 3-3 are given in Figure 5-3.

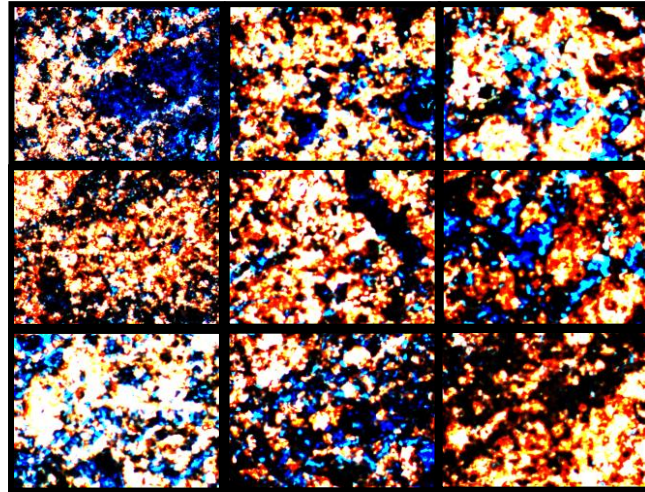


Figure 5-3: 3x3 imaging of first thin section of Core Plug 3-3

After imaging on all thin sections, analysis work had been carried out; ImageJ image analysis program had been used for this purpose. The

type of all taken images had been converted to the 8 bit images from its original 32 bit mode. Pixels on the images have been thresholded to have pore and particle views. As it can be seen in the Figure 5-4, after threshold adjustment, particles were represented in black and pore sections were represented in white.

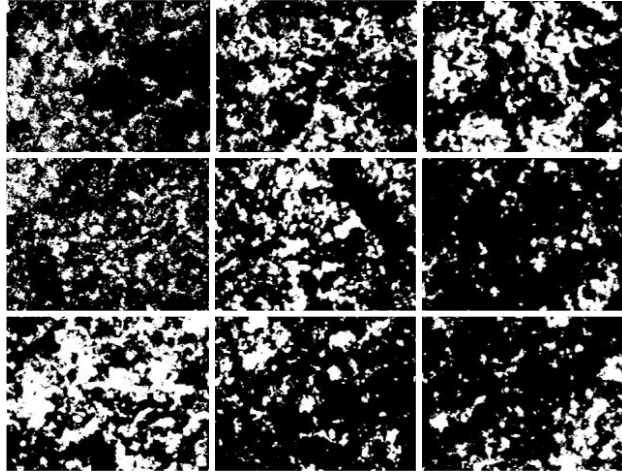


Figure 5-4: Representation of Pore and Particles for First Thin Section of Core Plug 3-3

The last step of image processing was analyzing the voids. By using the ImageJ program, areas of the pores have been obtained then the required radius data calculated accordingly. After that, data obtained from the separate images of 1 thin section had been amalgamated to make a statistical investigation.

In order to understand whether there is any correlation between the radii of two adjacent thin sections of same core plug or not, a statistical study has been performed. T-test on mean value of radii of two adjacent thin sections and in the case of unable to define difference, F-test on standard deviations, has been deployed. After those tests, it was observed that there is no correlation between two adjacent thin sections statistically. A sample set

of results on t-test has been given in Table 5-2 and Table 5-3. Table 5-4 reflects the results of F-test data analysis.

Table 5-2: Results of t-test on Thin Section #1 and #2 of Core Plug 3-3

	<i>Thin Section # 1</i>	<i>Thin Section # 2</i>
Mean	2.553705101	1.765358549
Variance	18.79368401	7.278000319
P	2.05441E-36	

The t-test analysis is the test for equality of the population means that underlie each sample. In this part of the study, radii of the pores, which have been obtained from the image analysis by using the ImageJ image analysis program, have been put into t-test in order to understand whether there is a correlation between the adjacent thin sections (imaged by using microscope with 10x magnification as mentioned before) of same core plug.

From the results of t-test on Thin Section # 1 and Thin Section # 2 of Plug 3-3, it can be said that the two means are significantly different, because Probability (P) is less than 0.05.

For the cases of having P greater than 0.05 as given in Table 5-3, there is a need to exert F-test on the same data set. Because of not having any statistical meaningful result in this case, F-test on standard deviations (variances) can give the required answer of whether the data sets are unique or coming from same global data set or not.

Table 5-3: Results of t-test on Thin Section #3 and #4 of Core Plug 3-3

	<i>Thin Section # 3</i>	<i>Thin Section # 4</i>
Mean	1.989474319	1.945869505
Variance	10.51421843	9.244570229
P	0.169726571	

Table 5-4: Results of F-test on Thin Section #3 and #4 of Core Plug 3-3

	<i>Thin Section # 3</i>	<i>Thin Section # 4</i>
Mean	1.989474319	1.945869505
Variance	10.51421843	9.244570229
P	3.53551E-10	

With the results of F-test, it can be said that the two standard deviations are significantly different ($P < 0.05$). As a result of the statistical analysis on the subjected radii data, it can be concluded that slice-wise radii data for all core plugs are not coming from the same global data set.

After completion of the statistical investigation on the pore radii data, slice-wise porosity has been calculated by using the pore and particle size areas obtained by the image analysis. Porosity distribution throughout the core plug 3-3 is given in Figure 5-5 (see the Appendix E for the other core plugs' porosity distributions; Figure E-4...Figure E-6)

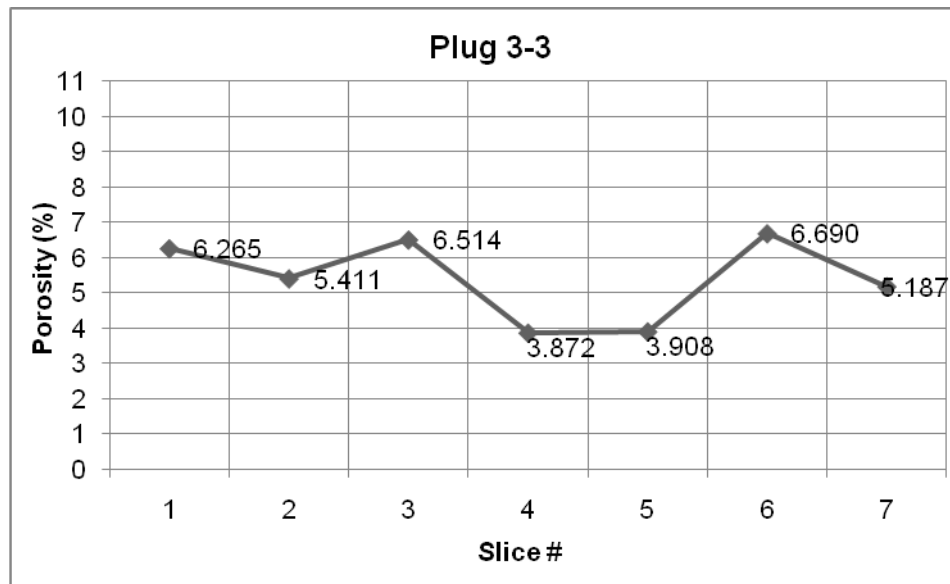


Figure 5-5: Porosity Distribution for Plug 3-3 from Thin Section Imaging

After that, pore size distribution study has been carried out on thin section imaging data (Figure 5-6 for core plug 3-3 and Figure E-7...Figure

E-9 for the rest). This study was the basis of developing the constructed pore network computer program.

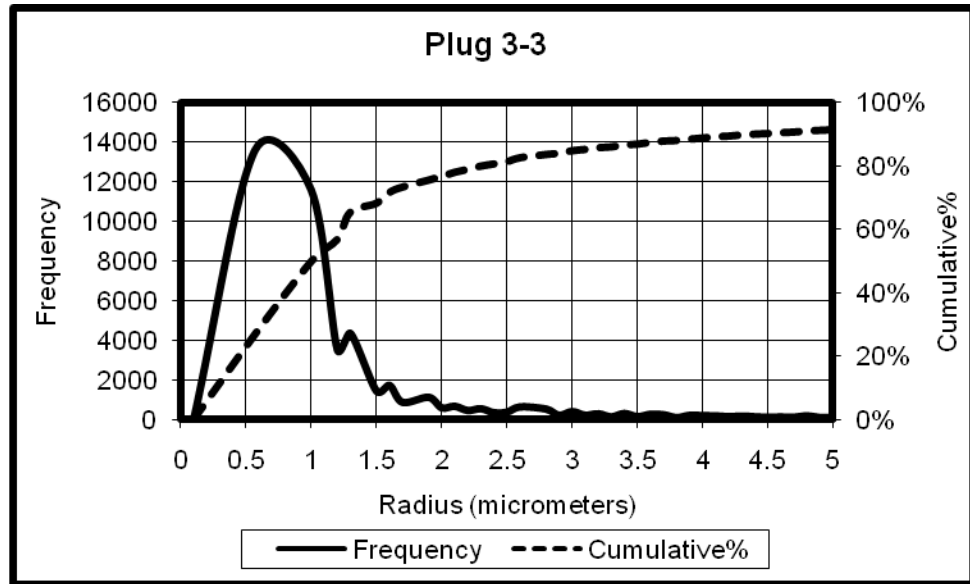


Figure 5-6: Frequency & Cumulative % vs. Radius (Plug 3-3)

In Fig. 5-6, the first concavity represents the pore throat number and the second the pore-body. By taking into account the ratio between pore-throat to pore-body, it was decided to keep coordination number of 4 as used in the first construction step of the pore network.

Assuming three dimensional pore network structure (8x8x8), which has cubic pore-bodies and pore-throats in shape and also assuming negligible volumes of pore-throats in saturation calculations, different displacement procedures were generated. The assumption of the ignorance of the volumes of the throats gave us the ability to discount the gravitational effects on the fluid displacements.

In order to test the correctness of the network model philosophy dynamic drainage and quasi-static imbibition procedures of different kinds of oil & water for the Weibull (Singh and Mohanty, 2003) distributed pore-bodies & pore-throats were simulated.

Pore-body and pore-throat radii distributions were described using a probability distribution function given below:

$$PDF(x) = \frac{x}{x_2^2} \exp\left(-\frac{x^2}{x_2^2}\right) \quad \text{for } x \leq x_3$$

and

$$= 0 \quad \text{for } x > x_3 \quad (5-2)$$

where $x = r - r_{\min}$, $x_2 = r_{ave} - r_{\min}$ and $x_3 = r_{\max} - r_{\min}$, r represents the pore-body or pore-throat radius, r_{\min} and r_{\max} are the minimum and maximum sizes respectively and r_{ave} is the average. As it can be seen from Eq. 5-2, r_{\min} , r_{\max} and r_{ave} are required to be defined in order to exert Weibull distribution into pore network as pore-body and pore-throat size distributions. In addition, θ (the contact angle) should be taken into account; and in this thesis, scope had been enlarged from perfectly water-wet media ($\theta = 0^\circ$) to perfectly oil-wet ($\theta = 180^\circ$). In order to cover mixed-wet situation, ninety degrees contact angle has been also used ($\theta = 90^\circ$). Table 5-5 lists the parameters for different pore-body and pore-throat size distributions which have been obtained by image analysis of subjected thin sections.

Table 5-5: Pore Network Model Radius Distribution Parameters of Core Plugs

<i>Core Plug #</i>	<i>Pore-body</i> $r_{\max}, r_{\min}, r_{ave}, \sigma$	<i>Pore-throat</i> $r_{\max}, r_{\min}, r_{ave}, \sigma$
3-3	1.492705, 1.261566, 1.318912, 0.09984	1.128379, 0.56419, 0.800182, 0.228568
4-1	1.492705, 1.261566, 1.290352, 0.076329	1.128379, 0.56419, 0.857946, 0.206875
92-2	1.492705, 1.261566, 1.297667, 0.083919	1.128379, 0.56419, 0.857388, 0.209824
100-4	1.492705, 1.261566, 1.325143, 0.103222	1.128379, 0.56419, 0.783147, 0.231766

While the distribution work had been carried out, it was seen that some radius were randomized in out of range. Then, a calculation patch that taking into account the standard deviation (σ) had been inserted to the pore network to keep radii in between the minimum and maximum values.

As pore-bodies and pore-throats have been chosen in square cross-section in shape, there was a need to insert zero roundedness coefficients into pore network model to depict the sharp corners. Conductance calculations of the fluids inside both of pore-bodies and –throats have been done by assuming the no slip boundary on solid – wetting phase contact line and no stress boundary on wetting – non-wetting phase contact. Model has been formed to be capable of handling the no slip boundary wetting – non-wetting phase contact line and also no stress boundary on solid – wetting phase contact line, too.

In order to be used in the probable future studies, pore network model was also enhanced by putting inside a patch of shape alteration mechanism. But, studies on shape alteration have been kept out of scope of this thesis.

The model is also capable of calculation of capillary pressures. The calculated capillary pressure data in three different contact angle cases are given in Figure 5-7. Not having the real capillary pressure data on subjected core plugs that have been used in this study, comparison study could not be carried out between the real and calculated data sets.

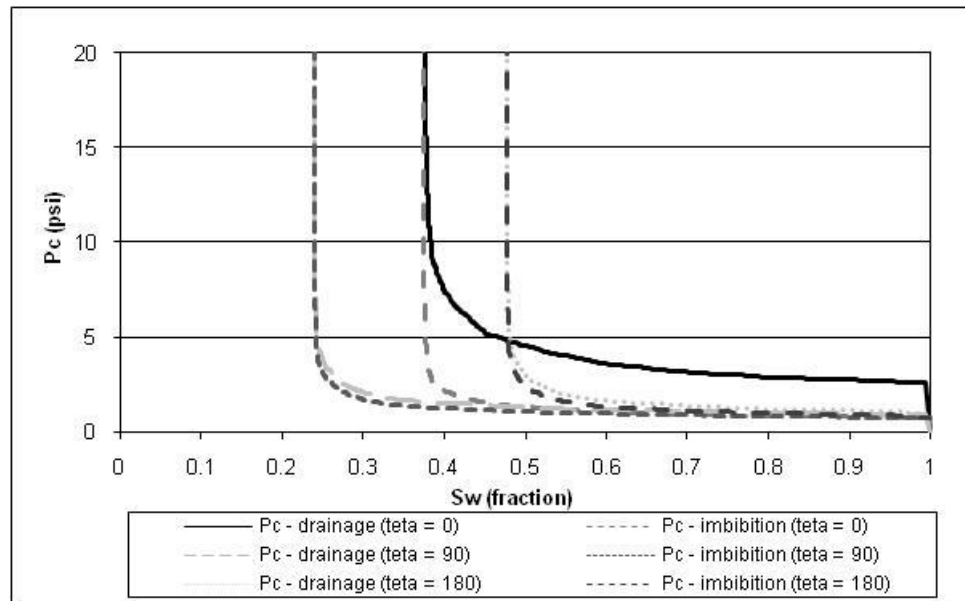


Figure 5-7: Calculated Capillary Pressures for Systems Having Different Contact Angles

Results gathered from the pore network model showed us that by choosing same contact angle throughout the system same irreducible water saturations were achieved at every run. Then, in order to make a comparison to test pore network model's two phase oil – water displacement and also relative permeability prediction efficiency, different contact angles have been tried to reach the same irreducible water saturation after drainage process with the one coming from the real experimental data sets.

Displacement simulations have been cut around 200 psi as the maximum capillary pressure for a real system having the same pore dimension. This assumption was taken in order to omit arising of any probable discrepancy between the real cases and the synthetic ones generated with pore network model. After having reached the plausible typical resultant output data from the developed computer program, the data were compared with the ones obtained from the laboratory experiments done on the core plugs taken from the real reservoirs (Comparison of

Experimental – “exp” data and the ones obtained from Pore Network – “PN” are given in Figure 5-8 and Figure 5-9).

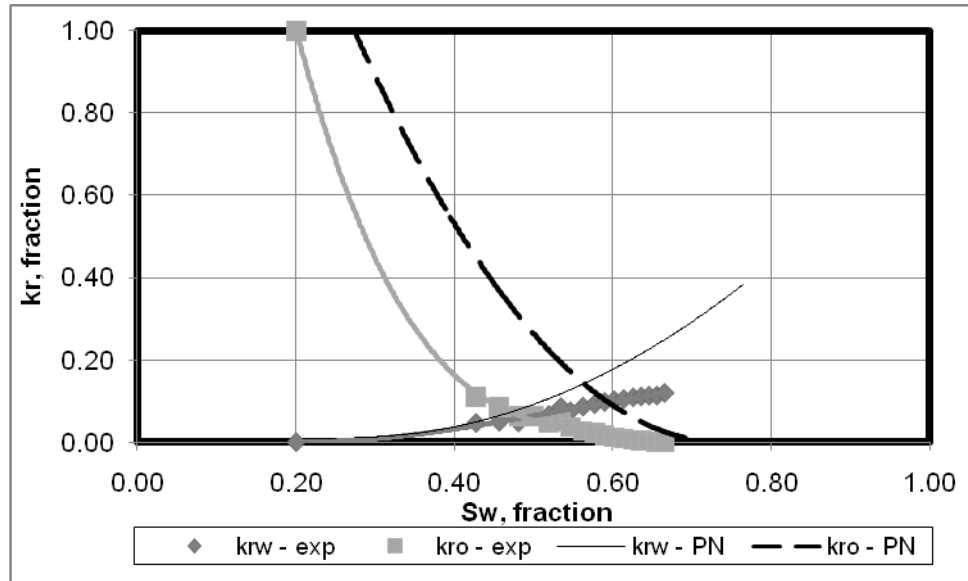


Figure 5-8: Relative Permeabilities for Plug 92-2 (before modification in calculations)

As mentioned before, in pore network model water – oil drainage simulations executed first. While this simulations, it was aimed to reach same irreducible water saturation at the end of the drainage that was the starting point of available real imbibition data sets. Those real experimental data sets have been taken from TPAO Research Center. In order to reach the aim, a study on contact angle has been done while network model simulations. After the subjected irreducible water saturation data has been reached, imbibition displacements have been simulated in the model with the same contact angle. The subjected data sets on core plug 92-2 and 100-4 are put into plots given in Figure 5-8 and 5-9 respectively.

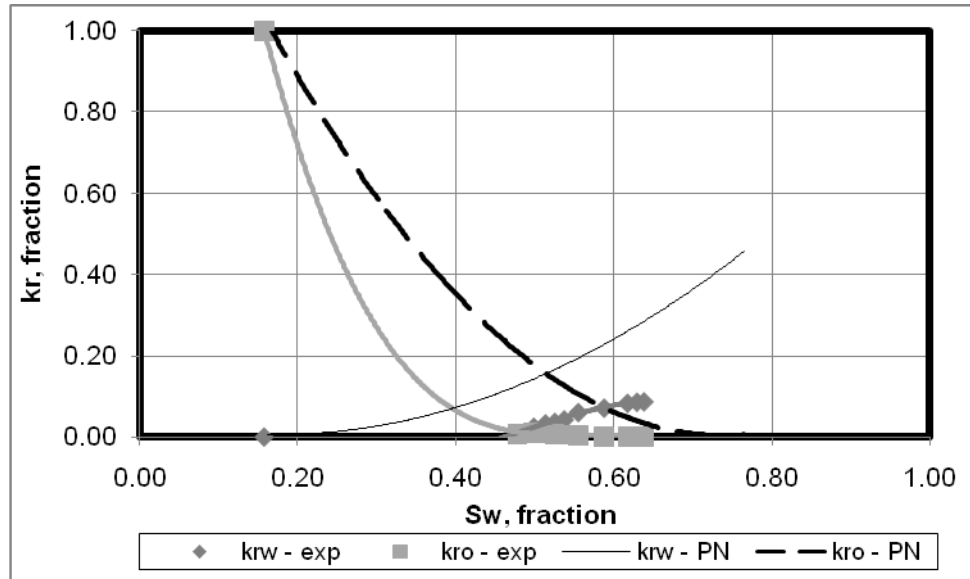


Figure 5-9: Relative Permeabilities for Plug 100-4 (before modification in calculations)

As it can be seen from the Fig. 5-8 and 5-9, the end points could not be matched by just only changing the contact angles. The reason for it could be having the small network size and capillary end effects, but unfortunately network limits could not be increased because of limited capacity of computer in use. In order to get better fits, network structure has not been changed but being in the decision of being affected from the capillary end effects of the model, first and the last 2 pore bodies in all directions of the model have not been put into consideration while doing conductance and also saturation calculations.

As it can be seen from Fig. 5-10 and 5-11, pore network model became to generate more accurate results after the subjected omission. Average error between the calculated (data from Pore Network – “PN”) and the real data (Experimental – “exp”) of water relative permeability for core plug 92-2 was found as 0.02282, and 0.02039 for oil relative permeability.

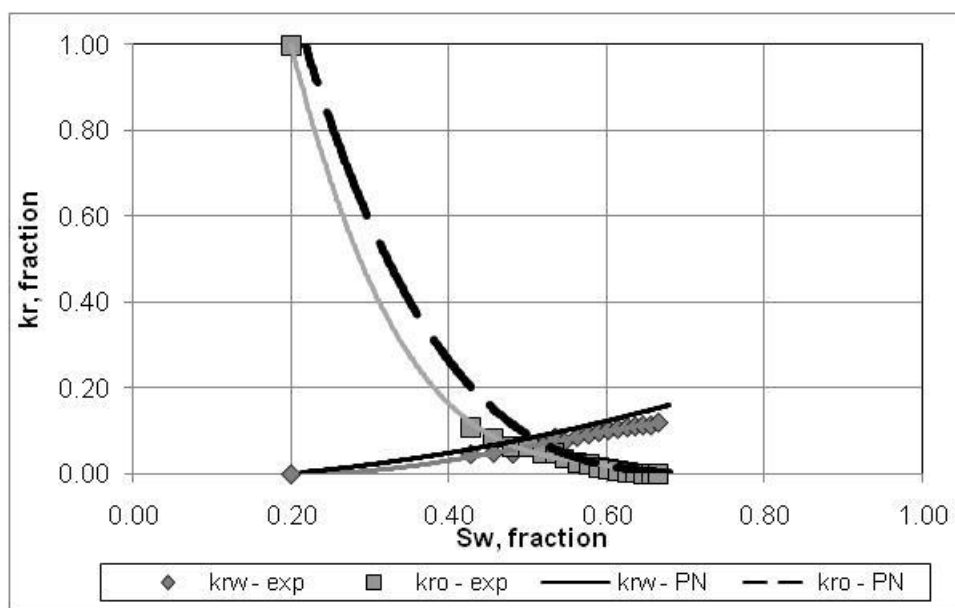


Figure 5-10: Relative Permeabilities for Plug 92-2 (after modification in calculations)

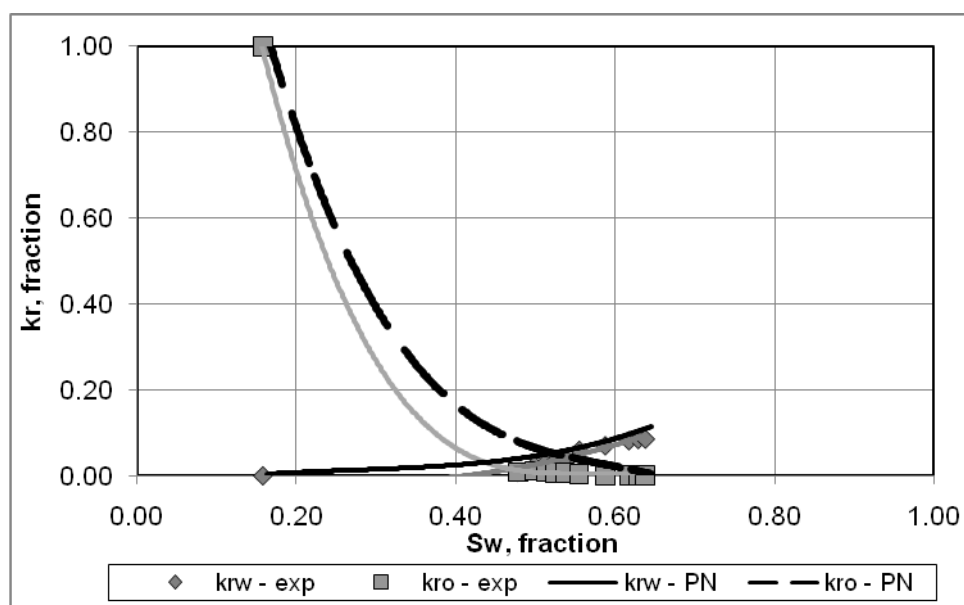


Figure 5-11: Relative Permeabilities for Plug 100-4 (after modification in calculations)

For the core plug 100-4, the average error between the water relative permeability data of pore network and of experimental one (depicted in Fig. 5-11) was found as 0.01609, and 0.03339 for the oil relative permeability.

Although the calculated relative permeability data of oil and water does not 100% fit (the possible reason of this could be, not having 100% matched pore-throat and pore-body distributions throughout the constructed pore network) with the real data, under the light of calculated average errors, it can be observed that the developed pore network model's prediction ability became more reliable. Being reached to this plausible predicted data was an encouraging step to look ahead of this study.

5.2. NMR Subroutine

At the last part of the pore network modelling, a NMR calculation patch has been added to developed network program, in order to calculate T2 relaxation times. With the addition of this procedure the tool has been enabled to generate relative permeability curves and T2 relaxation times for different rock-fluid compositions.

By choosing contact angle 0° (perfectly water-wet), 5 different temperatures (20, 40, 60, 80 and 100°C), corresponding water viscosities (1.003, 0.6531, 0.4668, 0.3550 and 0.2822 cp), oil API (10, 20, 30, 40 and 50 °API) and surface relaxivities for different core plug types (15 μs for sandstone, 1 μs for limestone, 5 μs for dolomitic limestone and 10 μs for dolostone), T2 relaxation times and relative permeability data have been produced by the network model.

In order to put viscosity effect of oil on T2 relaxation, the API gravity should be converted to it by using Eq. 5-3. For this, Modified Kartoatmodjo (Sattarin et. al., 2007) model has been used.

$$\mu_o = 220.15 \times 10^9 \times T^{(-3.5560)} \times \left[\log^{0.75} API \right]^{(12.5428 \times \log(T) - 45.7874)} \quad (5-3)$$

The obtained T2 relaxation times and relative permeability data from the pore network model were the input of first modelling step of neural network study. Then, in order to cover the mixed-wet and oil-wet situations 90° and 180° of contact angles have been added to data sets. Results of those two neural network development phases are given in the following chapter.

5.3. Geological Investigation on Core Plugs

In order to confirm rock types, a geological investigation on thin sections has also been carried out by Assoc. Prof. Dr. Bora ROJAY, Geological Engineering Dept., METU. Results of this study are given in Table 5-6.

Table 5-6: Geological Investigation Results on Core Plugs

<i>Core Plug #</i>	<i>Rock Type</i>	<i>Groundmass</i>	<i>Fragments</i>	<i>Porosity</i>	<i>Permeability</i>	<i>Oil</i>
4-1	Sandstone	Sparry Calcite	Quartz, less opaque minerals, no skeletal fragments, no fossils, no mold	40% ± 5	High (interlocked pores)	1-2 %
92-2	Dolostone	Microdolomite – Dolomite	Microsparite nodules	12% ± 5	Poor	no oil
3-3	Limestone	Sparry Calcite	Fossils (millioids, forams), skeletal fragments, molds, red alga	20% ± 5	Poor	no oil
100-4	Dolomitic Limestone	Sparry Calcite, Secondary Dolomitization	Skeletal fragments (2%), mold (1%)	10% ± 2	Poor	no oil

CHAPTER 6

IMPLEMENTATION TO THE NEURAL NETWORK MODELLING

In this part of the study an Artificial Neural Network (ANN) computer program was developed by using Turbo Pascal 7.0. The network type and the learning rule were chosen as fully connected and back-propagated error-correcting learning with momentum respectively. The correctness of this program was tested by using Exclusive-OR (XOR) function problem then it was recognized that the program traces correctly (see Appendix C for details).

After that, the program was educated by using the data obtained from the Pore Network (PN) model in order to obtain the requested two-phase oil – water relative permeability data for any given fluid configuration and also any given porous media type with known NMR values. Then, temperature, corresponding water viscosity, oil API, surface relaxivities, pore radii, water saturation and T2 data have been put into ANN as input layer data and water and oil relative permeabilities as output layer data in order to train the subjected ANN tool.

This study has two parts; first part takes into account the water-wet system (contact angle (θ) = 0) data; the latter water-wet, mixed-wet ($\theta = 90^\circ$) and oil-wet ($\theta = 180^\circ$). In order to teach ANN program to handle the data 100 different cases have been created by PN model (Figure 6-1) and one of them was tested after training process completed (Fig. 6-6 and 6-7). Throughout the sensitivity runs of ANN model, learning rate and momentum term were kept same and constant as 0.3 and 0.7 respectively for all cases given in the following sections.

6.1. Calculation of Error and Finding Global Minimum at Error Space

The error is the amount by which the predicted value differs from the quantity to be estimated and, the Mean Square Error (MSE) measures the average of the square of the error.

Error on this work refers to difference between desired value and the calculated output. After obtaining error in every data set, average of the square of the error (i.e. MSE) has been calculated. For one iteration step, those MSE values have been added up and compared with the desired error value that has been declared before by ANN model user. After that, this calculated total error has been propagated backwards, if the comparison does not meet the criteria (i.e. calculated total MSE should be less than or equal to desired error), to all nodes to change the weight values of every unit. Then with the new weights, forward runs have been implemented to calculate new output.

These iteration steps have been carried out till to reach the stopping criteria mentioned above. Philosophy under doing this is reaching the Least Mean Square Error (LMSE). By this way, data calculated by ANN would be the best fit on desired values.

6.2. Sensitivity Study on ANN Model While Learning With Data of Water-wet Systems

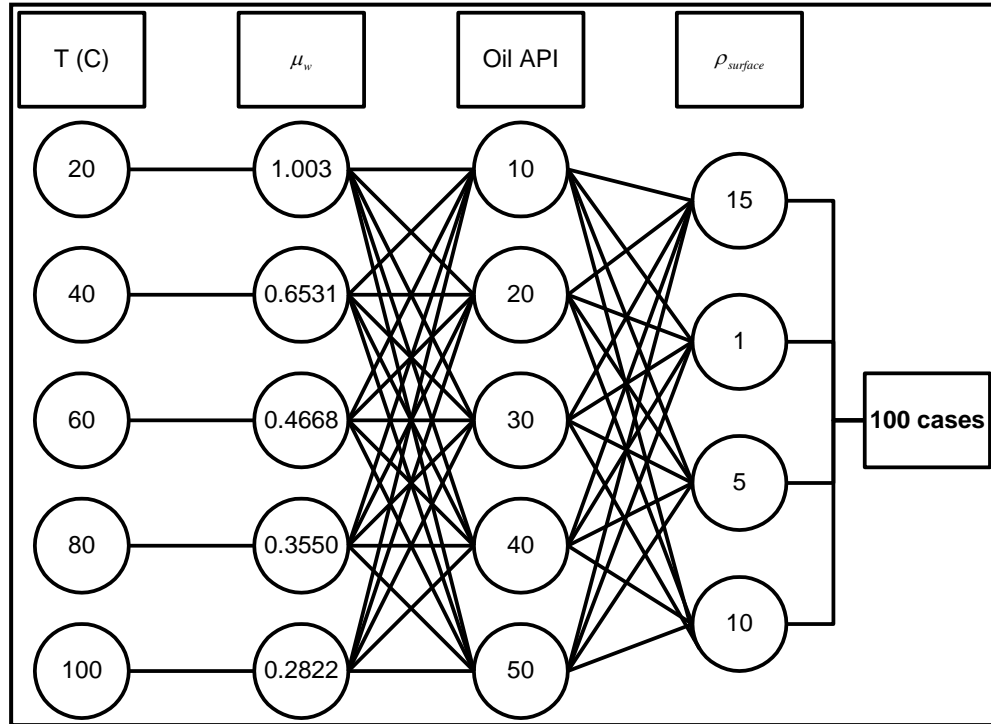


Figure 6-1: Generated Data Set by Pore Network Model for Part # 1

At the first step, it was decided to test the basic architecture of ANN which is having input layer, 1 hidden layer with 5 nodes and an output layer (Figure 6-2). After 52,000 iterations (epoch), LMSE of 7.89 had been reached.

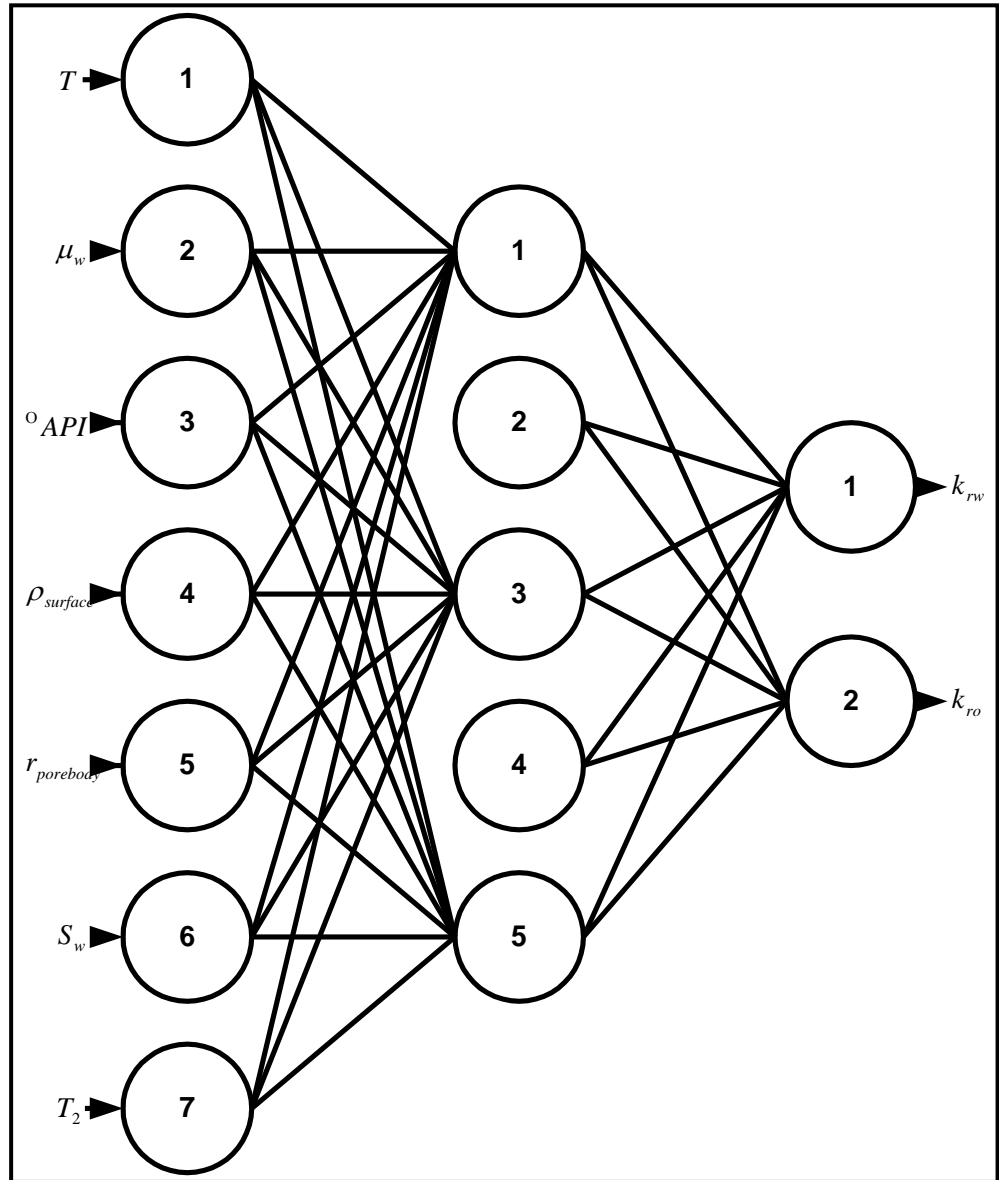


Figure 6-2: Schematic diagram of two-phase relative permeability model
(Stage#1)

At the second step, under the light of the results of first step, it was decided to use 2 hidden layers with 5 and 3 nodes respectively (Figure 6-3). After 42,000 epochs, LMSE of 3.94 had been reached.

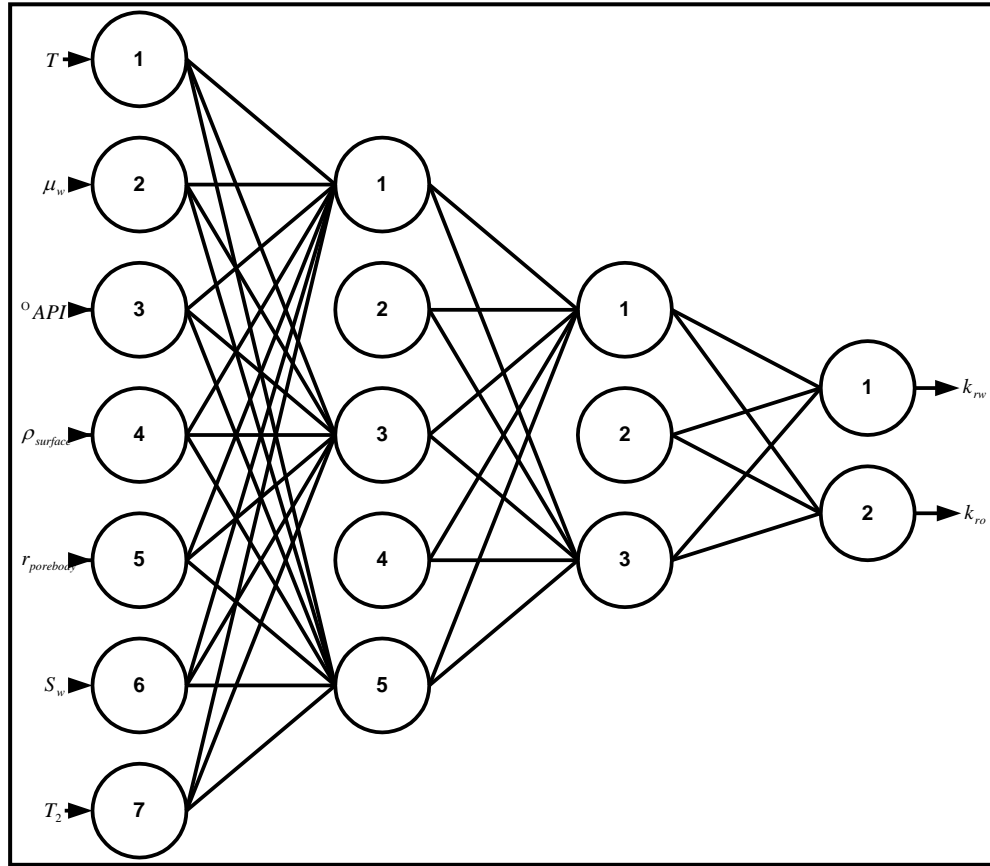


Figure 6-3: Schematic diagram of two-phase relative permeability model
(Stage#2)

At the third step, it was decided to use three hidden layers (with 7, 5 and 3 hidden nodes respectively, Figure 6-4) and seen that network reached LMSE of 3.08 at the 20,000 epoch; but it increased to 3.66 at the 46,310 epoch and training had been stopped because of increase in the trend.

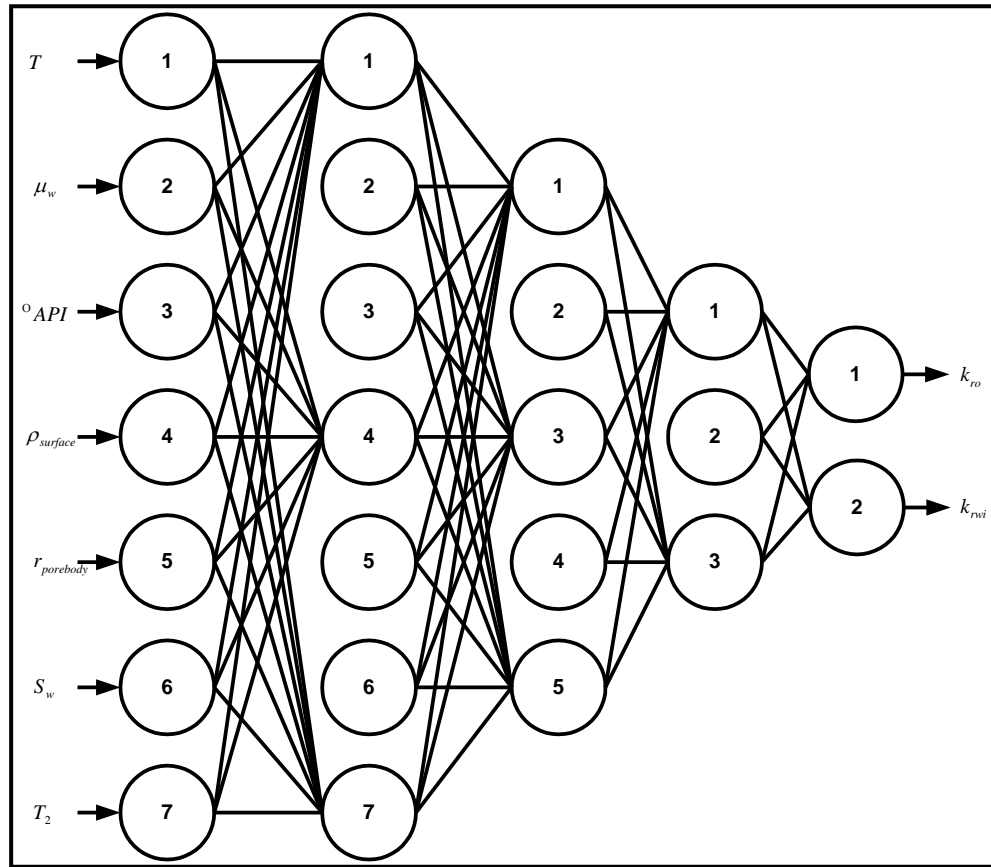


Figure 6-4: Schematic diagram of two-phase relative permeability model
(Stage#3)

At the last stage, it was decided to run two ANN tools for water and oil relative permeabilities individually. Hidden layer structure (7, 5 and 3 nodes at respective layers) has been kept same as used in stage#3 (Figure 6-5).

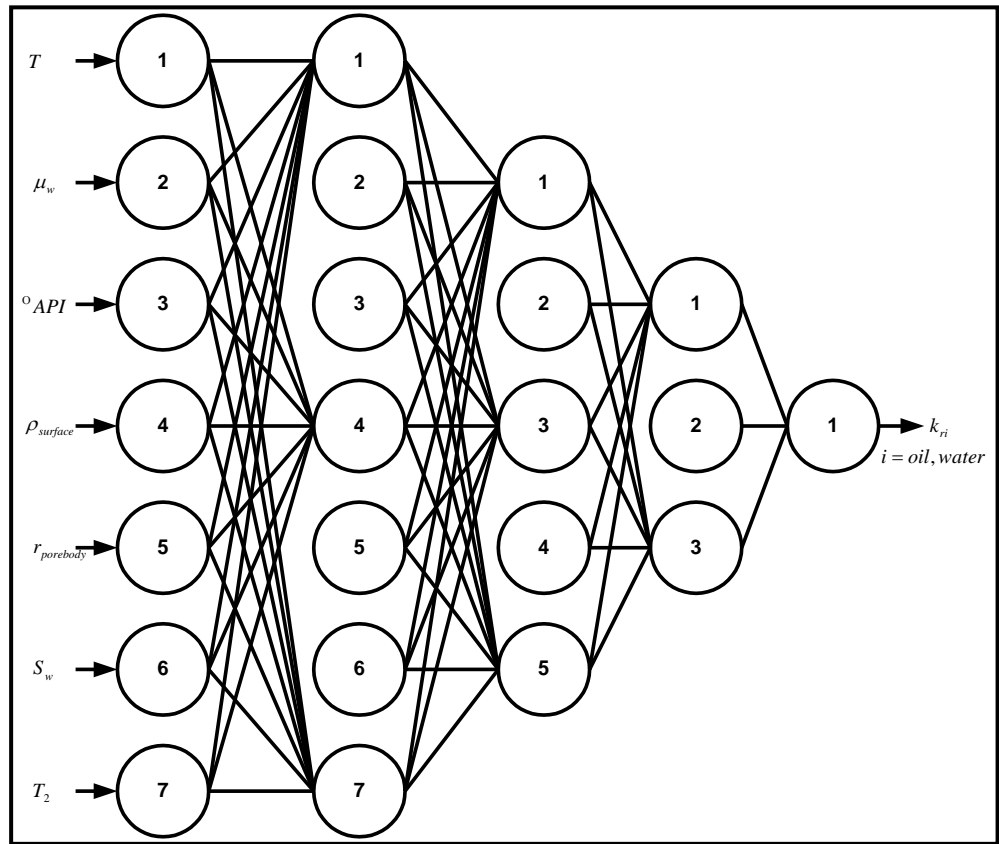


Figure 6-5: Schematic diagram of two-phase relative permeability model
(Stage#4)

After the ninth iteration, oil relative permeability predictor has reached the desired minimum LMSE of 0.01 and program stopped itself. The reason for reaching the desired error immediately could be having the oil relative permeability data lies between zero to 0.00706.

Water relative permeability predictor has reached the LMSE of 1.3576 at the 49,800 epoch and around 49,900 iteration it started to increase till to peak of 1.6109; then started to decrease again. It was seen that LMSE of 1.3576 cannot be decreased more or even reached again because of reaching the oscillation in error space, it is decided to stop water relative permeability predictor and use the corresponding weight vector of this 1.3576 LMSE.

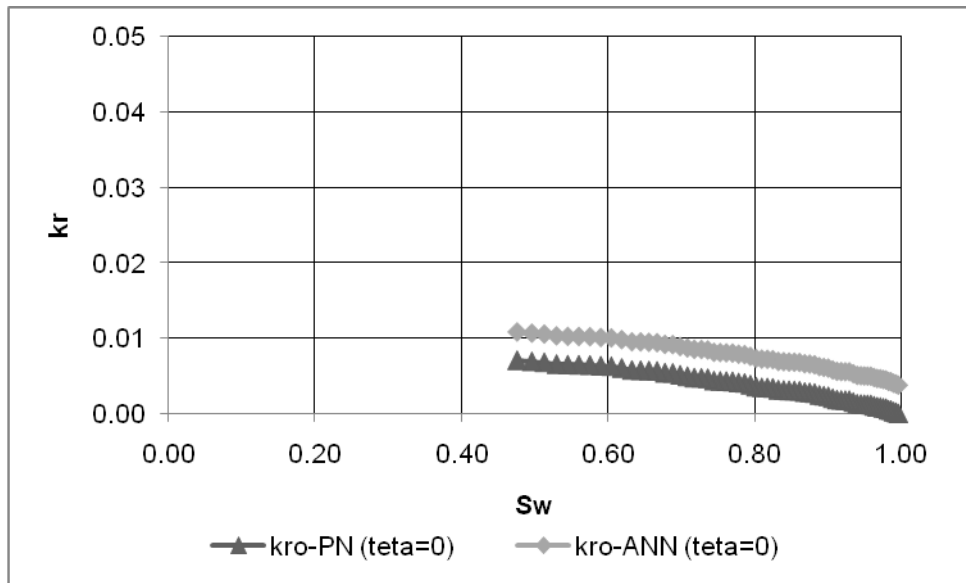


Figure 6-6: Comparison of Oil Relative Permeabilities Obtained From PN and ANN (Contact Angle = 0°)

Comparison graphs plotted with the data of oil and water relative permeability predictors and the real one are given in Figure 6-6 and Figure 6-7 for 0° contact angle situation.

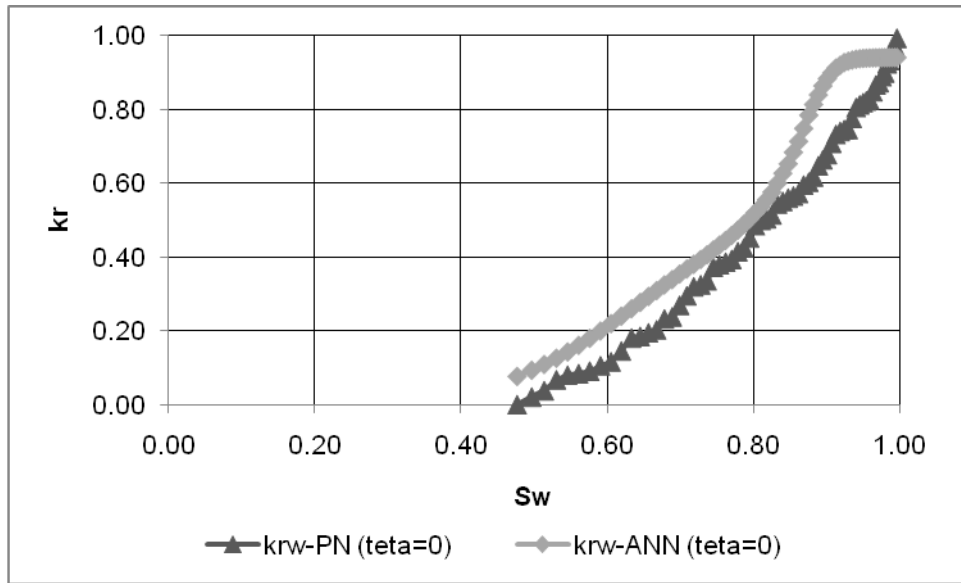


Figure 6-7: Comparison of Water Relative Permeabilities Obtained From PN and ANN (Contact Angle = 0°)

It can be seen from Fig. 6-7 that water relative permeability predictor has good estimations from 0.476 to 0.8 water saturations; and after the value of 0.8, predictions becomes out of trend of pore network estimated water relative permeability curve. Instead of reaching top value (initial stage of drainage), which is relative permeability of 1 at 100% water saturation, neural network estimations become concentrated at 0.94 for water saturations from 0.935 to 0.997. For the zero water relative permeability at 47.6% water saturation, neural network has estimated the value as 0.0784. This shift between estimated and pore network data might be the result of not permitting neural network learning process long enough to continue its iterations to reach 0.01 desired LMSE value.

It can be observed from the network structures, hidden layer node (or unit) numbers were chosen either same as input layer or lesser. It is obvious that the time should be spent to reach the desired LMSE will increase with increasing node numbers, because in every iteration step, system should take

into account also those additional nodes (and their weights) in the calculations. So, reason of decreasing node numbers in the structure was trying to decrease learning time.

6.3. Enlargement of the Span of the Training Data

With having plausible results from the constructed ANN model for water-wet porous media, it was decided also to cover mixed and oil-wet systems. Without changing the last network structure tried at Stage # 4, the data for learning stage had been enlarged with 90° and 180° (in addition to 0°) contact angles in order to focus on all wettability situations. Figure 6-8 shows the created cases by using developed pore network model.

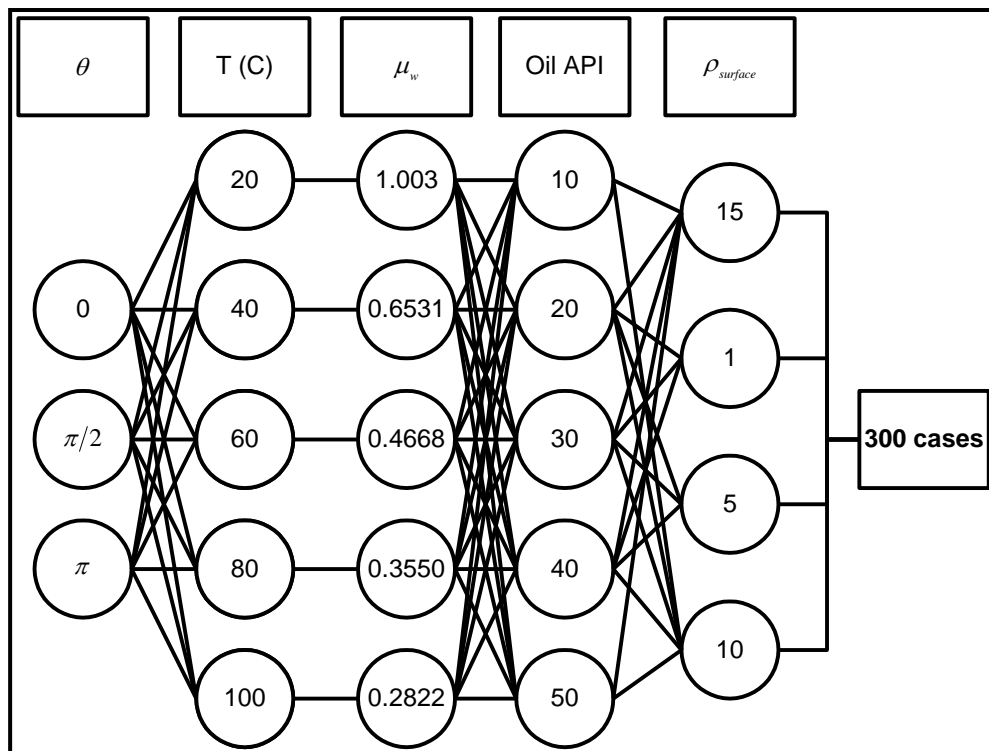


Figure 6-8: Generated Data Set by Pore Network Model for Part # 2

Both of the ANN predictors had been stopped at around 38,000 iterations, because the learning data sets were huge and the calculated

LMSE at each iteration was not becoming less as fast as at the beginning (change in error values between two adjacent iteration step was around 10^{-6}).

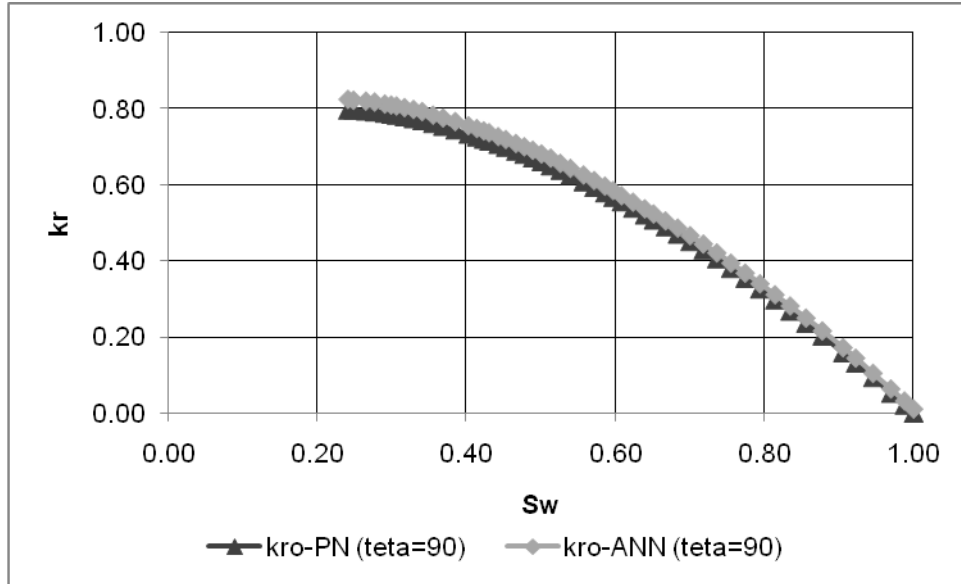


Figure 6-9: Comparison of Oil Relative Permeabilities Obtained From PN and ANN (Contact Angle = 90°)

Figure 6-9 and 6-10 depict the comparison between the oil and water relative permeability data of mixed-wet system obtained by using pore and neural network models respectively. As it can be seen from the figures oil relative permeabilities predicted by ANN almost perfectly fit with the ones obtained from PN. LMSEs have been found as 0.92047 and 2.18046 for oil and water relative permeability data respectively. It can also be said that while the slope of the curve changes, LMSE between the predicted and tested data increases. But we have the better estimates for oil relative permeabilities than water counter parts. This situation brings the question in minds that whether the predictions have been affected by concavity of training data or not. It can be hypothesized by looking into both figures, Figure 6-9 and 6-10, that predictions are affected by concavity (i.e. the

correlation between saturation and relative permeability) either up or down of the training data sets.

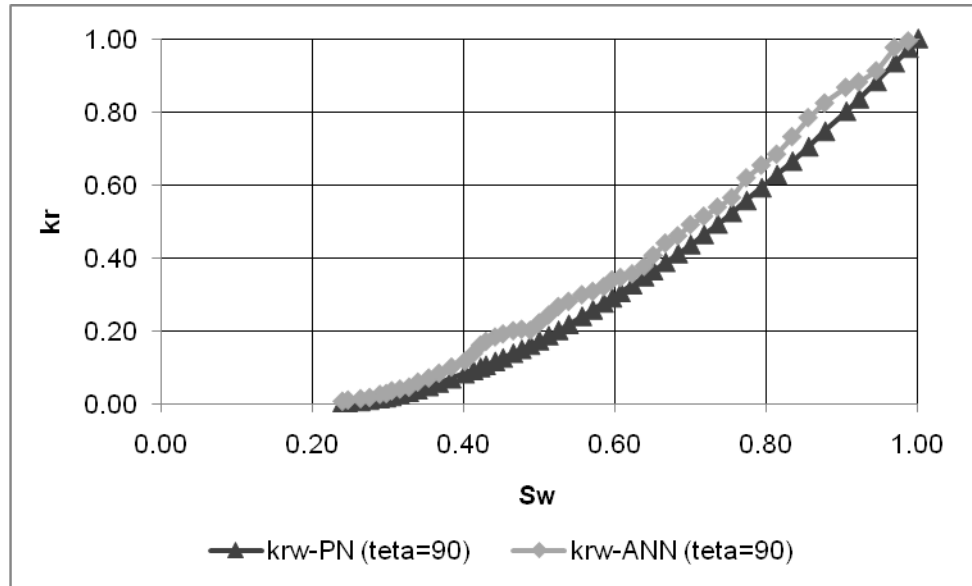


Figure 6-10: Comparison of Water Relative Permeabilities Obtained From PN and ANN (Contact Angle = 90°)

While we look at the Figure 6-11 and 6-12 which show comparison between oil and water predicted and test data of relative permeability versus water saturation for oil-wet media, it can be said that hypothesis given above becomes a conclusion as, the prediction of ANN is highly affected by relative permeability distribution with saturation changes. This conclusion can be much more strengthened by analyzing the Figure 6-12. Like in Fig. 6-10, this figure shows us that at every big change in slope at test data (while the concave down situations) the predicted data obtained from ANN show sudden increases. That could be the reason of having small concave up figures in predicted data sets of water relative permeabilities.

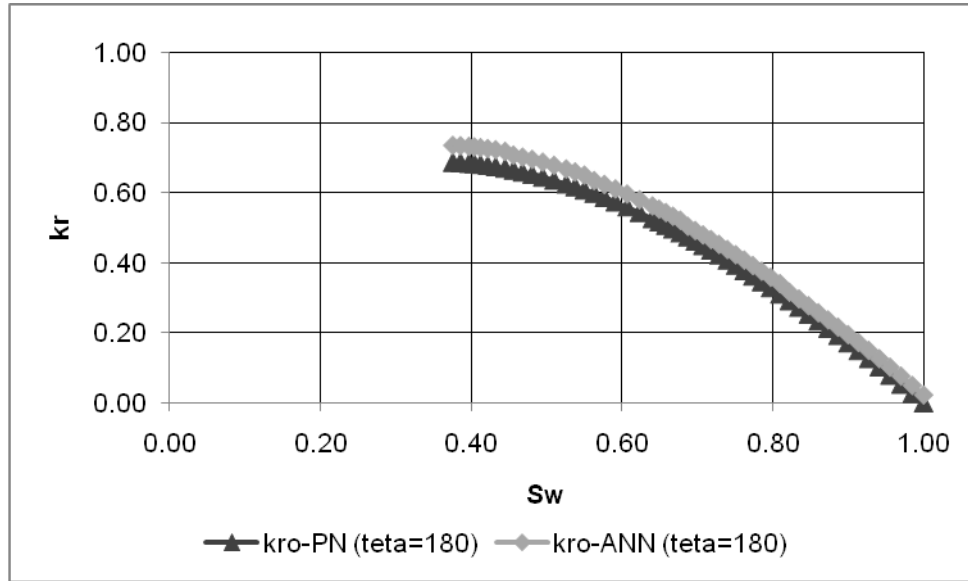


Figure 6-11: Comparison of Oil Relative Permeabilities Obtained From PN and ANN (Contact Angle = 180°)

Error calculations show that the LMSE is 1.68532 for oil relative permeabilities, and 2.70545 for the water relative permeabilities between predicted and testing data sets, which are a little bit higher than learning errors.

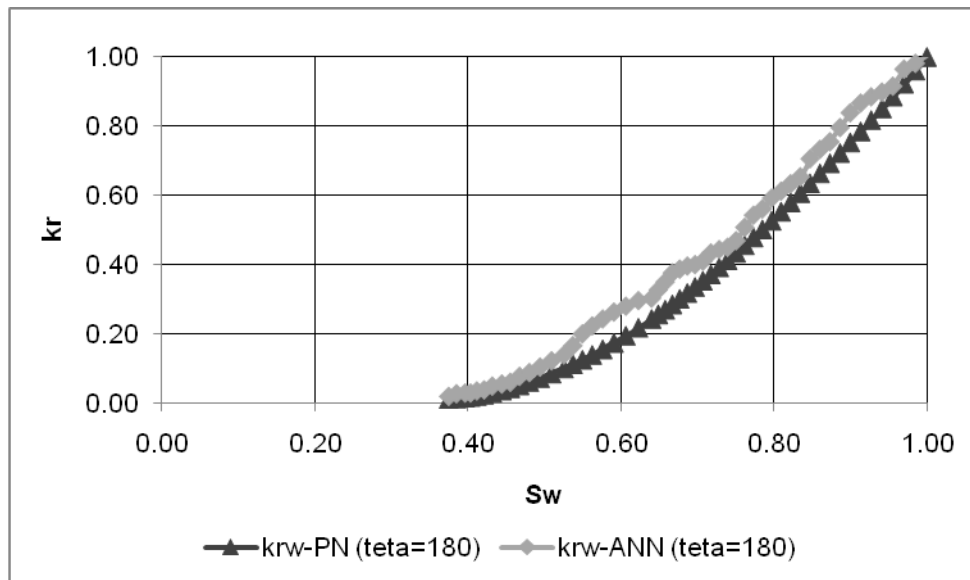


Figure 6-12: Comparison of Water Relative Permeabilities Obtained From PN and ANN (Contact Angle = 180°)

CHAPTER 7

CONCLUSIONS

An Artificial Neural Network (ANN) model trained using data obtained from a Pore Network (PN) model was developed for estimating relative permeabilities and capillary pressures. With the developed tools two phase oil-water relative permeability data for different wettabilities can be easily calculated by using NMR T2 relaxation data that can be obtained from well logging. The following observations and conclusions can be drawn based on the results obtained from constructed PN model where the wettability can be easily converted from fully water-wet to mixed-wet and also to oil-wet by choosing the contact angle between 0 and 180° and the ANN model:

1. The developed PN model generated realistic relative permeability and capillary pressure data avoiding time and money consuming coring operations and experimental displacement works.
2. Statistical study carried out using the successive thin section image data reflected that pore radii data were not statistically correlated.

3. PN model can be run with different pore shapes.
4. Same irreducible water saturations were obtained for every core plug by choosing a unique contact angle throughout the system. However, the end point relative permeabilities could not be matched by simply changing the contact angles. In order to obtain better matches capillary end effects were minimized by omitting first and the last pore layers in all conductance and saturation calculations. It is possible to obtain a better match by accurately matching experimental pore-throat and pore-body distributions.
5. The developed pore network successfully generates relative permeability curves and T2 relaxation times for different rock-fluid compositions.
6. Geological study performed on thin sections has showed that exact rock typing is needed while choosing corresponding surface relaxivities.
7. Using momentum term in the ANN model decreased the iterations.
8. Sensitivity study on network architecture has shown that the best ANN has 3 hidden layers with 7, 5 and 3 nodes respectively. It is also observed that the use of separate ANN's for oil and water relative permeability data decreased time spent on reaching the required minimum error criteria.
9. While training the ANN model with oil relative permeability data of zero contact angle case, learning was completed after 9 iterations. The reason for reaching the desired error

immediately could be the low end point oil relative permeability.

10. It is observed that water relative permeability predictions were better for low saturations. The difference between PN and ANN data might due to insufficient number of iterations.
11. Results of the mixed-wet case showed that oil relative permeabilities predicted by ANN almost perfectly fit with the ones obtained from PN.

CHAPTER 8

RECOMMENDATIONS

In this study, a Pore Network model that simulates the two phase displacements and a Neural Network model that predicts the two phase oil-water relative permeabilities for different rock and fluid properties were developed. This study can be enhanced by giving more time to the training procedure of the artificial neural network. Secondly, rate effects may be handled more efficiently by running the network with different shape pores thus resulting in better relative permeability estimations.

Real-time NMR logging empowered by the constructed artificial neural network model that uses the synthetic data generated by developed pore network model, can become more robust if a dynamical mode by restarting learning phase of ANN with the addition of the new field data sets is used.

REFERENCES

- Agut, R., Levallois, B., Klopff, W., (2000): "Integrating Core Measurements and NMR Logs in Complex Lithology", SPE 63211, Proceedings of the 2000 SPE Annual Technical Conference and Exhibition, pp. 1-14.
- Akin, S., Demiral, B., Okandan. E., (1996): "A Novel Method of Porosity Measurement Utilizing Computerized Tomography", In Situ, Vol. 20, No. 4, pp. 347 – 365.
- Arns, J. Y., Arns, C. H., Sheppard, A. P., Sok, R. M., Knackstedt, M. A., Pinczewski, W. V., (2003): "Relative Permeability from Tomographic Images; Effect of Correlated Heterogeneity", Journal of Petroleum Science and Engineering, 39, pp. 247 – 259.
- Bakke, S., Oren, P.E., (1996): "3-D Pore-Scale Modeling of Heterogeneous Sandstone Reservoir Rocks and Quantitative Analysis of the Architecture, Geometry and Spatial Continuity of the Pore Network", SPE 35479, Proceedings of the SPE European 3-D Reservoir Modeling Conference, pp. 35 – 45.
- Berkowitz, B., Balberg, I., (1992): "Percolation Approach to the Problem of Hydraulic Conductivity in Porous Media", Transport in Porous Media, 9(3), pp. 275 – 286.

- Billiotte, J. A, Moegen, H. D., Oren, P., (1990): “Experimental Micromodeling and Numerical Simulation of Gas/Water Injection/Withdrawal Cycles as Applied to Underground Gas Storage”, SPE 20765, Proceedings of the 1990 SPE Annual Technical Conference and Exhibition, pp. 483 – 491.
- Bobroff, S., Guillot, G., Riviere, C., Cuiec, L., Roussel, J. C., Kassab, G., (1995): “Microscopic Arrangement of Oil and Water in Sandstone by NMR Relaxation Times and NMR Imaging”, J. Chim Phys, 92, pp. 1885 – 1889.
- Bowers, M. C., Ehrlich, R., Howard, J. J., Kenyon, W. E., (1995): “Determination of Porosity Types from NMR Data and Their Relationship to Porosity Types Derived from Thin Section”, Journal of Petroleum Science and Engineering, vol. 13, pp. 1 – 14.
- Broadbent, S. R., Hammersley, J. M., (1957): “Percolation Processes”, Proc. Cambridge Philos. Soc., 53, pp. 629 – 645.
- Carmen, P.C., (1937): “Fluid Flow Through Granular Beds”, Transactions of the Institution of Chemical Engineers, 15, pp.150 – 160.
- Carr, M. B., Ehrlich, R., Bowers, M. C., Howard, J. J., (1996): “Correlation of Porosity Types Derived From NMR Data and Thin Section Image Analysis in a Carbonate Reservoir”, Journal of Petroleum Science and Engineering, 14, pp. 115 – 131.
- Celia, M.A., Reeves, P.C., (1995): “Recent Advances in Pore Scale Models for Multiphase Flow in Porous Media”, U.S. National Report to IUGG, Rev. Geophys. Vol. 33 Suppl.

- Cory, D. G., Reichwein, A. M., van OS J. W. M., Veeman, W. S., (1988): “NMR Images of Rigid Solids”, Chemical Physics Letters, Vol. 143, Number 5, pp. 467 – 470.
- Davies, D.K., Vessell, R.K., (1995): “Flow Unit Characterization of a Shallow shelf Carbonate Reservoir: North Robertson Unit, West Texas”, SPE/DOE 35433, Proceedings of the 1995 SPE/DOE Tenth Symposium on Improved Oil Recovery, pp. 295 – 304.
- Dixit, A.B., McDougall, S.R., Sorbia, K.S., Buckley, J.S., (1996): “Pore Scale Modeling of Wettability Effects and Their Influence on Oil Recovery”, SPE 35451, Proceedings of the 1996 SPE/DOE Tenth Symposium on Improved Oil Recovery, pp. 501 – 515.
- Dullien, F.A.L., (1991): “Characterization of Porous Media: Pore Level”, Transport in Porous Media, 6, pp. 581 – 606.
- Fatt, I., (1956a): “The Network Model of Porous Media, I. Capillary Pressure Characteristics”, Trans. AIME, Vol:207, pp. 144 – 159.
- Fatt, I., (1956b): “The Network Model of Porous Media, II. Dynamic Properties of a Single Size Tube Network”, Trans. AIME, Vol:207, pp. 160 – 163.
- Fatt, I., (1956c): “The Network Model of Porous Media, III. Dynamic Properties of Networks with Tube Radius Distribution”, Trans. AIME, Vol:207, pp. 164 – 181.
- Fenwick, D.H., Blunt, M.J., (1997): “Use of Network Modeling to Predict Saturation Paths, Relative Permeabilities and Oil Recovery for Three Phase Flow in Porous Media”, SPE 38881, Proceedings of the 1997 SPE Annual Technical Conference and Exhibition, pp. 383 – 396.

- Freedman, R., Morriss, C. E., (1995): "Processing of Data from an NMR Logging Tool", SPE 30560, Proceedings of the 1995 SPE Annual Technical Conference & Exhibition, pp. 301 – 316.
- Fu, L., (1994): "Neural Networks in Computer Intelligence", McGraw-Hill, Inc., Singapore.
- Haines, W.B., (1930): "The Hysteresis Effect in Capillary Properties and the Modes of Moisture Distribution", Studies in the physical properties of soil 5, J. Agric. Sci., 20, pp. 97 – 116.
- Haykin, S., (1994): "Neural Networks", Macmillan College Publishing Inc., New York.
- Hecht-Nilsen, R., (1989): "Neurocomputing", Addison-Wesley, Reading, Mass.
- Hertz, J., Krogh, A., Palmer, R. G., (1991): "Introduction to the Theory of Neural Computation", Perseus Books Publishing, L.L.C., Massachusetts.
- Honarpour, M. M., Koederitz, S. M., (1986): "Relative Permeability of Petroleum Reservoirs", CRC Press, Inc., Boca Raton, FL.
- Hui, Mun-Hong, Blunt, M.J., (2000): "Pore-Scale Modeling of Three-Phase Flow and the Effects of Wettability", SPE 59309, SPE/DOE Improved Oil Recovery Symposium.
- Ioannidis, M., Kwiecien, M., Chatzis, I., (1995): "Computer Generation and Application of 3-D Model Porous Media: From Pore-Level Geostatistics to the Estimation of Formation Factor", SPE 30201, Proceedings of the 1995 SPE Petroleum Computer Conference, Houston, pp. 185 – 194.

- Kamath, J., Xu, B., Lee, S. H., Yortsos, Y. C., (1996): "Pore Network Modeling of Laboratory Experiments on Heterogeneous Carbonates", SPE 36681, Proceedings of the 1996 SPE Annual Technical Conference and Exhibition, Denver, pp. 61 – 67.
- Kantzas, A., Chatsiz, I., Dullien, F.A.L., (1998): "Mechanisms of Capillary Displacement of Residual Oil by Gravity-Assisted Inert Gas Injection", SPE 17506, Proceedings of the 1998 SPE Rocky Mountain Regional Meeting, Casper, pp. 297 – 307.
- Karaman, T., Demiral, B.M.R., (Aug 2004): "Determination of Three Phase Relative Permeability Values by Using an Artificial Neural Network Model", Energy Sources, Vol. 26, Num. 10, pp. 903-914.
- Karaman, T., Demiral, B.M.R., (June 2004): "Determination of Two and Three Phase Relative Permeability Values by Using a Pore Network Model", Energy Sources, Vol. 26, Num. 7, pp. 685-696.
- Karaman, T., Demiral, B., (2003a): "Two and Three Phase Relative Permeability Estimations by Using Pore Network Modeling", Proceedings of the 14th International Petroleum and Natural Gas Congress and Exhibition of Turkiye, Ankara, Turkiye, pp. 322-332.
- Karaman, T., Demiral, B., (2003b): "Three Phase Relative Permeability Estimation by Using Artificial Neural Networks", Proceedings of the 14th International Petroleum and Natural Gas Congress and Exhibition of Turkiye, Ankara, Turkiye, pp. 333-343.
- Karaman, T., Erdogan, I., Demiral, B., (1998): "Relative Permeability Estimation by using Pore Network Modeling", Proceedings of the 12th International Petroleum and Natural Gas Congress and Exhibition of Turkiye, 1998, Ankara, Turkiye, pp. 671-686.

- Kharabaf, H., Yortsos, Y.C., (1996): “A Pore-Network Model for Foam Formation and Propagation in Porous Media”, SPE 36663, Proceedings of the 1996 SPE Annual Technical Conference and Exhibition, Denver, pp. 779 – 790.
- Kleinberg, R. L., Straley, C., Kenyon, W. E., Akkurt, R., Farooqui, S. A., (1993): “Nuclear Magnetic Resonance of Rocks: T1 vs. T2”, SPE 26470, Proceedings of the 68th SPE Annual Technical Conference and Exhibition, Houston, Texas, pp. 553 – 563.
- Koplik, J., Lasseeter, T.J., (1985): “Two-Phase Flow in Random Network Models of Porous Media”, SPE 11014, Society of Petroleum Engineers Journal, pp. 89 – 100.
- Lee, S.H., Padmanabhan, L, Al-Sunaidi, H.A., (1995): “Simulation of Linear Displacement Experiments on Massively Parallel Computers”, SPE 30721, Proceedings of the 1995 SPE Annual Technical Conference and Exhibition, Dallas, pp. 191 – 204.
- Lenormand, R., Roland, Z., Zarcone, C., Cesar, L., (1984): “Role of Roughness and Edges During Capillary Imbibition in Square Capillaries”, SPE 13264, Proceedings of the 1984 SPE 59th Annual Technical Conference and Exhibition, Houston.
- Lizhi, X., Youru, D., Chaohui, Y., (1995): “Experimental Study on NMR Spectra of Fluids in the Pores of Rock”, Science in China (Series A), Vol. 38, No. 11, pp. 1333 – 1343.
- Lo, S. W., Hirasaki, G. J., House, W. V., Kobayashi, R., (2000): “Mixing Rules and Correlations of NMR Relaxation Time With Viscosity, Diffusivity, and Gas/Oil Ratio of Methane/Hydrocarbon Mixtures”, SPE 77264, Proceedings of the 2000 SPE Annual Technical Conference and Exhibition, Dallas, pp. 24 – 34.

- Lu, D., Zhou, M., Dunsmuir, J. H., Thomann, H., (2001): “NMR T2 Distributions and Two Phase Flow Simulations from X-ray Micro-Tomography Images of Sandstones”, Magnetic Resonance Imaging, 19, pp. 443 – 448.
- Mani, V., Mohanty, K.K., (1997): “Pore-Level Network Modeling of Three-Phase Capillary Pressure and Relative Permeability Curves”, SPE 38883, Proceedings of the 1997 SPE Annual Technical Conference and Exhibition, San Antonio, Texas, pp.405 – 418.
- McDougall, S.R., Sorbie, K.S., (1995): “The Impact of Wettability on Waterflooding: Pore-Scale Simulation”, SPE 25271, SPE Reservoir Engineering , pp. 208 – 213.
- Mogensen, K., Stenby, E., (1998): “A Dynamic Pore-Scale Model of Imbibition”, SPE 39658, Proceedings of the 1998 SPE/DOE Improved Oil Recovery Symposium , Tulsa, Oklahoma, pp. 48 – 59.
- Narayanan, K., Deans, H.A., (1988): “A Flow Model Based on the Structure of Heterogeneous Porous Media”, SPE 15328, Proceedings of the 1988 SPE 63rd Annual Technical Conference and Exhibition, Houston, pp. 676 – 690.
- Nilsen, L. S., Oren, P. E., Bakke, S., Henriques, A., (1996): “Prediction of Relative Permeability and Capillary Pressure from a Pore Model”, SPE 35531, Proceedings of the 1996 SPE European 3-D Reservoir Modeling Conference, Norway, pp. 345 – 354.
- Nguyen, V. H., Sheppard, A. P., Knackstedt, M.A., Pinczewski, W. V., (2004): “A Dynamic Network Model for Imbibition”, SPE 90365, Proceedings of the 2004 SPE International Petroleum Conference in Puebla, Mexico, pp. 1 – 21.

- Parra, J. O., Hackert, C. L., Collier, H. A., Bennet, M., (2001): “NMR and Acoustic Signatures in Vuggy Carbonate Aquifers”, Proceedings of the 42nd Annual SPWLA Meeting, Houston, Texas, pp. 1 – 12.
- Paterson, L., Painter, S., Zhang, X., Pinczewski, V., (1996): “Simulating Residual Saturation and Relative Permeability in Heterogeneous Formations”, SPE 36523, Proceedings of the 1996 SPE Annual Technical Conference and Exhibition, Denver, pp. 321 – 328.
- Paterson, L., Lee, J.Y., Pinczewski, W.V., (1997): “Three-Phase Relative Permeability in Heterogeneous Formations”, SPE 38882, Proceedings of the 1997 SPE Annual Technical Conference and Exhibition, San Antonio, Texas, pp. 397 – 403.
- Patzek, T. W., (2000): “Verification of a Complete Pore Network Simulator of Drainage and Imbibition”, SPE 59312, Proceedings of the 2000 SPE / DOE Improved Oil Recovery Symposium, Tulsa, Oklahoma, pp. 1 – 15.
- Roberts, A.P., Bentz, D.P., Knackstedt, M. A., (1996): “Correlating Microstructure to the Petrophysical Properties of Porous Rocks”, SPE 37024, Proceedings of the 1996 SPE Asia Pacific Oil and Gas Conference, Australia, pp. 551 – 559.
- Rojay, B. (2008): Personal Communication for Geological Investigation on Core Plugs.
- Satik, C., Yortsos, Y.C., (1995): “Pore-Network Studies of Steam Injection In Porous Media”, SPE 30751, Proceedings of the 1995 SPE Annual Technical Conference and Exhibition, Dallas, pp. 1 – 15.

- Sattarin, M., Modarresi, H., Bayat, M., Teymori, M., (2007): “New Viscosity Correlations for Dead Crude Oils”, *Petroleum & Coal*, 49(2), pp. 33 – 39.
- Silpngarmlers, N., Guler, B., Ertekin, T., Grader, A.S., (2001): “Development and Testing of Two Phase Relative Permeability Predictors Using Artificial Neural Networks”, SPE 69392, Proceedings of the SPE Latin American and Caribbean Petroleum Engineering Conference, Buenos Aires, Argentina.
- Singh, M., Mohanty, K.K., (2003): “Dynamic Modeling of Drainage Through Three-dimensional Porous Materials”, *Chemical Engineering Science*, 58, pp. 1 – 18.
- van As, H., van Dusschoten, D., (1997): “NMR Methods for Imaging of Transport Processes in Micro-Porous Systems”, *Geoderma*, vol. 80, pp. 389 – 403.
- van Brakel, J., (1975): “Pore Space Models for Transport Phenomena in Porous Media: Review and Evaluation with Special Emphasis on Capillary Liquid Transport”, *Powder Technology*, 11, pp. 205 – 236.
- Washburn, E.W., (1921): “The Dynamics of Capillary Flow”, *Phys. Rev.*, 17, pp. 273 – 283.
- Watts, J.W., (1971): “An Iterative Matrix Solution Method Suitable for Anisotropic Problems”, SPE 2802, SPEJ.
- Watts, J.W., (1973): ”A Method for Improving Line Successive Overrelaxation in Anisotropic Problems – A Theoretical Analysis”, SPE 3559, SPEJ.
- Wenrong, M., Shihong, S., Tianhua, L., Wenzhong, L., Guoheng, H., (1996): “Pore and Throat Network Model and Its Application to

Optimal Selection of Temporary Plugging Particles”, SPE 31099, Proceedings of the 1996 SPE International Symposium on Formation Damage, Louisiana, pp. 293 – 300.

Xu, B., Kamath, J. Yortsos, Y.C. and Lee, S. H., (1997): “Use of Pore Network Models of Simulate Laboratory Corefloods in a Heterogeneous Carbonate Sample”, SPE 38879, Proceedings of the 1997 SPE Annual Technical Conference and Exhibition, San Antonio, pp. 359 – 367.

APPENDIX A

SOURCE CODE OF PORE NETWORK MODELLING

```
Program Pore;
uses crt;
Label back;
Const
  ArrSize = 7;
  SigmaOW = 50/1000; {N/m} {0.048,0.05}
  SigmaOG = 24/1000; {N/m} {0.019,0.24}
  Len      = 100; {pore-throat length}
  ncorners = 4; {Corners of pore-body & pore-throat}
  Teta     = 0*pi/180; {contact angle}
  rd       = 0; {roundedness, rd=0 if the corner is sharp}
  f1       = 1; {solid/wetting phase contact line, f1=1 if no slip boundary}
  f2       = 0; {wetting/nonwetting fluid boundary, f2=1 for no slip
boundary, f2=0 for no-stress boundary}
  MuW      = 1.0;
  MuO      = 10.0;
  rpbodymin = 1.261566;
  rpbodymax = 1.492705;
  rpbodyave = 1.318912;
  rpbodystddev = 0.09984;
  rpthroatmin = 0.56419;
  rpthroatmax = 1.128379;
  rpthroatave = 0.800182;
  rpthroatstddev= 0.228568;
  error     = 0.1; {error in pressure relaxation in pores}
  Pinlet    = 1000; {inlet pressure set}
type
  DataType = Record
    Rpbody,
    Gpbodynw,Gpbodyw,
    Pcpbody,Pctemp,
    Giiw,Giinw,
```

```

        Gjjw,Gjjnw,
        Gkkw,Gkknw,
{*****      indexpb,      *****}
        P,Pnw,Pw,
        Gpb1fnw,Gpb1fw,
        Gii1fw,Gii1fnw,
        Gjj1fw,Gjj1fnw,
        Gkk1fw,Gkk1fnw,
        Q1fnw,Q1fw,Qw,Qnw,
        krnw,krw      : Real;
        End;

DataType2    = Record
        Rpthroati,Rpthroatj,Rpthroatk,
        Gpthroatinw,Gpthroatjnw,Gpthroatknw,
        Gpthroatiw,Gpthroatjw,Gpthroatkw,
        Pcpthroati,Pcpthroatj,Pcpthroatk,
{*****      indexpthroati,indexpthroatj,indexpthroatk,      *****}
        Gpti1fnw,Gptj1fnw,Gptk1fnw,
        Gpti1fw,Gptj1fw,Gptk1fw      : Real;
        End;

PointDataType = ^DataType;
PointDataType2 = ^DataType2;
        MainArrayType = Array [0..ArrSize,0..ArrSize,0..ArrSize] Of
PointDataType;
        MainArrayType2 = Array [0..ArrSize+1,0..ArrSize+1,0..ArrSize+1]
Of PointDataType2;

var
relative:    MainArrayType;
relative2:   MainArrayType2;
i,j,k,ii,jj,kk:   integer;
alfa:    real;
Lengt: text;

{*****}
Procedure Initialize_1;
Var i,j,k : Integer;
Begin
For i:=0 To ArrSize+1 Do
For j:=0 to ArrSize+1 Do
For k:=0 To ArrSize+1 Do
Begin
        Relative[i,j,k]^Rpboddy:=0;

```

```

        Relative[i,j,k]^Gpbodynw:=0;
        Relative[i,j,k]^Gpbodyw:=0;
        Relative[i,j,k]^Pcpbody:=0;
        Relative[i,j,k]^Giiw:=0;
        Relative[i,j,k]^Giinw:=0;
        Relative[i,j,k]^Gjjw:=0;
        Relative[i,j,k]^Gjjnw:=0;
        Relative[i,j,k]^Gkkw:=0;
        Relative[i,j,k]^Gkknw:=0;
{*****      Relative[i,j,k]^indexpb:=0;      *****}
        Relative[i,j,k]^Gpb1fnw:=0;
        Relative[i,j,k]^Gpb1fw:=0;
        Relative[i,j,k]^Qnw:=0;
        Relative[i,j,k]^Qw:=0;
        Relative[i,j,k]^Q1fnw:=0;
        Relative[i,j,k]^Q1fw:=0;
        Relative[i,j,k]^Pctemp:=0;

    End;
    For i:=0 To ArrSize+1 Do
    For j:=0 to ArrSize+1 Do
    For k:=0 To ArrSize+1 Do
    Begin
        Relative[i,j,k]^P:=0;
        Relative[i,j,k]^Pnw:=0;
        Relative[i,j,k]^Pw:=0;
    end;
    For i:=0 To ArrSize+1 Do
    For j:=0 to ArrSize+1 Do
    For k:=0 To ArrSize+1 Do
    Begin
        Relative2[i,j,k]^Rpthroati:=0;
        Relative2[i,j,k]^Rpthroatj:=0;
        Relative2[i,j,k]^Rpthroatk:=0;
        Relative2[i,j,k]^Gpthroatinw:=0;
        Relative2[i,j,k]^Gpthroatjnw:=0;
        Relative2[i,j,k]^Gpthroatknw:=0;
        Relative2[i,j,k]^Gpthroatiw:=0;
        Relative2[i,j,k]^Gpthroatjw:=0;
        Relative2[i,j,k]^Gpthroatkw:=0;
        Relative2[i,j,k]^Pcpthroati:=0;
        Relative2[i,j,k]^Pcpthroatj:=0;
        Relative2[i,j,k]^Pcpthroatk:=0;
{*****      Relative2[i,j,k]^indexpthroati:=0;
        Relative2[i,j,k]^indexpthroatj:=0;

```

```

        Relative2[i,j,k]^indexpthroatk:=0;      **** }
Relative2[i,j,k]^Gpt1fnw:=0;
Relative2[i,j,k]^Gptj1fnw:=0;
Relative2[i,j,k]^Gptk1fnw:=0;
        Relative2[i,j,k]^Gpt1fw:=0;
        Relative2[i,j,k]^Gptj1fw:=0;
        Relative2[i,j,k]^Gptk1fw:=0;
    End;
End;
{ **** }
Function AritMean(Item1,Item2:Real):Real;
Begin
    AritMean:=(Item1+Item2)/2;
End;
{ **** }
Function HarMean(Item1,Item2:Real):Real;
Begin
    HarMean:=1/((1/2)*( 1/Item1) + (1/Item2) ));
End;
{ **** }
Function Log(xxx:Real):Real;
Begin
    Log:=Ln(xxx)/Ln(10);
End;
{ **** }
Function Power(xx:Real; j:Real):Real;
Begin
    If xx<0 Then
        If Odd(Trunc(j)) Then
            Power := (-1)*exp(j*ln(Abs(xx)))
        Else
            Power := exp(j*ln(Abs(xx)))
        Else If xx>0 Then
            Power := exp(j*ln(xx))
        Else If xx=0 Then
            Power:=0;
    End;
{ **** }
{ Wetting Phase Area }
Function Areaw(rwnw:Real):Real;
var angle,cot,cosqu,cosin: real;
Begin
    angle:=alfa+teta;
    cot:=cos(alfa)/sin(alfa);
    cosqu:=(1+cos(2*angle))/2;

```

```

    cossin:=cos(angle)*sin(angle);
    Areaw :=ncorners*Power(rwnw,2)*(cot*cossqu-(Pi/2-angle)+cossin);
End;
{ **** }
{ Nonwetting phase conductance in a pore-throat or a pore-body }
Function Gnw(r:Real;L:Real):Real;
var Anw,reff: real;
Begin
    Anw:=power(r,2)-Areaw(r);
    reff:=0.5*power(Anw/Pi+r,0.5);
    Gnw:=Pi*Power(reff,4)/(8*MuO*L);
End;
{ **** }
{ Wetting phase conductance in a pore-throat or a pore-body }
Function Gw(r:Real;L:Real):Real;
var
    Aw,Tav1,Tav2,Tav3,B,Beta,angle,cot,tan,cossqu,cossin,sinsqu,u,Betanumer
ator,Betadenominator: real;
Begin
    Aw:=Areaw(r);
    angle:=alfa+teta;
    cot:=cos(alfa)/sin(alfa);
    tan:=1/cot;
    cossqu:=(1+cos(2*angle))/2;
    cossin:=cos(angle)*sin(angle);
    sinsqu:=(1-cos(2*angle))/2;
    Tav1:=cossqu+cossin*tan;
    Tav2:=1-Teta/(Pi/2-alfa);
    Tav3:=cos(angle)/cos(alfa);
    B:=(Pi/2-alfa)*tan;
    Betanumerator:=12*sinsqu*power((1-B),2)*(Tav1-
B*Tav2)*power((Tav3+f2*B*Tav2-rd*(1-f1*B)),2);
    Betadenominator:=power((1-sin(alfa)),2)*power(B,2)*power((Tav1-
B*Tav2-(1-B)*power(rd,2)),3);
    Beta:=Betanumerator/Betadenominator;
    Gw:=Aw*power(r,2)/(Beta*MuW*L);
End;
{ **** }
{ ****Capillary Pressure in pore-bodies & pore-throats**** }
Function Pcpbpt(r:real):real;
var Pcpbptnumerator,Pcpbptdenominator,
    cossqu,cossin: real;
Begin

    cossqu:=(1+cos(2*teta))/2;

```

```

cossin:=cos(teta)*sin(teta);
Pcpbptnumerator:=(SigmaOW/r)*(teta+cossqu-Pi/4-cossin);
Pcpbptdenominator:=cos(teta)-Power((Pi/4-teta+cossin),0.5);
Pcpbpt:=Pcpbptnumerator/Pcpbptdenominator;
end;

{*****}
{*****Heaviside Function to Calculate Step Conductances*****}
Function H(x:real):real;
var kkk:real;
Begin
    if x>=0 then H:=1.0
    else H:=0.0;
end;
{*****}
Procedure AllocateArray_1;
Var i,j,k : Integer;
Begin
    For i:=0 To ArrSize Do
        For j:=0 To ArrSize Do
            For k:=0 To ArrSize Do
                Begin
                    New(Relative[i,j,k]);
                End;
            For i:=0 To ArrSize+1 Do
                For j:=0 To ArrSize+1 Do
                    For k:=0 To ArrSize+1 Do
                        Begin
                            New(Relative2[i,j,k]);
                        End;
                    End;
                End;
            End;
        End;
    End;
{*****}
Procedure DisposeArray_1;
Var i,j,k : Integer;
Begin
    For i:=0 To ArrSize Do
        For j:=0 To ArrSize Do
            For k:=0 To ArrSize Do
                Begin
                    Dispose(Relative[i,j,k]);
                End;
            End;
        End;
    End;
    For i:=0 To ArrSize+1 Do
        For j:=0 To ArrSize+1 Do
            For k:=0 To ArrSize+1 Do

```

```

Begin
Dispose(Relative2[i,j,k]);
End;

End;
{ **** }
Procedure PoreSizeDistribution;
label back,back2,back3,back4,back5,back6,back7,back8,
      back9,back10,back11,back12,back13,back14,back15,back16,
      back17,back18,back19,back20;
var r,gofx,func,u : Real;
    pro,iter : longint;
    Rpbsize,Rpysizei,Rpysizej,Rpysizek : text;
Begin
{ Pore-Body Size Distribution }
Assign(Rpbsize,'C:\Rpbsize.txt');
Rewrite(Rpbsize);
clrscr;
iter:=0;
For i:=1 to ArrSize Do
  For j:=1 to ArrSize Do
    For k:=1 to ArrSize Do
      Begin
back:
u:=random(10000)/10000.1;
r:=rpbodymin*(power(-Ln(u),1/rpbodymax));
if (r<rpbodymin) then
  begin
    r:=r+10*rpbodystddev;
  end
  else if (r>rpbodymax) then r:=r-10*rpbodystddev;
iter:=iter+1;
if (iter>1000000) then goto back2;
if (r<rpbodymin) then goto back;
back2:
ii:=random(ArrSize);
jj:=random(ArrSize);
kk:=random(ArrSize);
if (Relative[ii,jj,kk]^Rpbody=0) then
  begin
    Relative[ii,jj,kk]^Rpbody:=r;
  end
  else
  begin
    goto back2;
  end
      end
    end
  end
end

```



```

                                end;
end;
i:=ArrSize;
For j:=1 to ArrSize Do
  For k:=1 to ArrSize Do
    Begin
      u:=random(10000)/10000.1;
      r:=rpbodymin*(power(-Ln(u),1/rpbodymax));
      if (r<rpbodymin) then
        begin
          r:=r+10*rpbodystddev;
        end
      else if (r>rpbodymax) then r:=r-10*rpbodystddev;
      Relative[i,j,k]^Rpbody:=r;
    end;
  j:=ArrSize;
  For i:=1 to ArrSize Do
    For k:=1 to ArrSize Do
      Begin
        u:=random(10000)/10000.1;
        r:=rpbodymin*(power(-Ln(u),1/rpbodymax));
        if (r<rpbodymin) then
          begin
            r:=r+10*rpbodystddev;
          end
        else if (r>rpbodymax) then r:=r-10*rpbodystddev;
        Relative[i,j,k]^Rpbody:=r;
      end;
    k:=ArrSize;
    For i:=1 to ArrSize Do
      For j:=1 to ArrSize Do
        Begin
          u:=random(10000)/10000.1;
          r:=rpbodymin*(power(-Ln(u),1/rpbodymax));
          if (r<rpbodymin) then
            begin
              r:=r+10*rpbodystddev;
            end
          else if (r>rpbodymax) then r:=r-10*rpbodystddev;
          Relative[i,j,k]^Rpbody:=r;
        end;
      j:=0;
      For j:=1 to ArrSize Do
        For k:=1 to ArrSize Do
          Begin

```

```

Relative[i,j,k]^Rpbody:=0;
end;
j:=0;
For i:=1 to ArrSize Do
  For k:=1 to ArrSize Do
    Begin
      Relative[i,j,k]^Rpbody:=0;
      end;
    k:=0;
    For i:=1 to ArrSize Do
      For j:=1 to ArrSize Do
        Begin
          Relative[i,j,k]^Rpbody:=0;
          end;
        For ii:=1 to ArrSize Do
          For jj:=1 to ArrSize Do
            For kk:=1 to ArrSize Do
              Begin
                writeln(Rpbody,ii,' ',jj,' ',kk,"Relative[ii,jj,kk]^Rpbody);
              end;
            clrscr;
            writeln('Step#1.0 - Pore-body size distribution: ok');
            { ***** }
            { Pore-Throat Size Distribution }
            Assign(Rpbody,'C:\Rpbody.txt');
            Rewrite(Rpbody);
            Assign(Rpbodyj,'C:\Rpbodyj.txt');
            Rewrite(Rpbodyj);
            Assign(Rpbodyk,'C:\Rpbodyk.txt');
            Rewrite(Rpbodyk);
            iter:=0;
            For k:=1 to ArrSize Do
              For j:=1 to ArrSize Do
                For i:=1 to ArrSize+1 Do
                  Begin
                    back3:
                    u:=random(10000)/10000.1;
                    r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                    if (r<rpthroatmin) then
                      begin
                        r:=r+10*rpthroatstddev;
                      end
                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
                    iter:=iter+1;
                    if (iter>1000000) then goto back4;

```

```

if (r<rpthroatmin) then goto back3;
back4:
ii:=random(ArrSize+1);
jj:=random(ArrSize);
kk:=random(ArrSize);
if (Relative2[ii,jj,kk]^Rpthroati=0) then
    begin
        Relative2[ii,jj,kk]^Rpthroati:=abs(r);
    end
else
    begin
        goto back4;
    end;
end;
i:=ArrSize+1;
For j:=1 to ArrSize Do
    For k:=1 to ArrSize Do
        Begin
u:=random(10000)/10000.1;
r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroati:=abs(r);
end;
j:=ArrSize;
For i:=1 to ArrSize+1 Do
    For k:=1 to ArrSize Do
        Begin
u:=random(10000)/10000.1;
r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroati:=abs(r);
end;
k:=ArrSize;
For i:=1 to ArrSize+1 Do
    For j:=1 to ArrSize Do
        Begin

```

```

u:=random(10000)/10000.1;
r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroati:=abs(r);
end;
i:=0;
For j:=1 to ArrSize Do
    For k:=1 to ArrSize Do
        Begin
            Relative2[i,j,k]^Rpthroati:=0;
        end;
        {i*****}
For i:=1 to ArrSize+1 Do
    For j:=1 to ArrSize Do
        For k:=1 to ArrSize Do
            Begin
                back5:
                if (i=1) then
                    begin
                        if (Relative2[i,j,k]^Rpthroati>Relative[i,j,k]^Rpbody) then
                            begin
                                u:=random(10000)/10000.1;
                                r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                                if (r<rpthroatmin) then
                                    begin
                                        r:=r+10*rpthroatstddev;
                                    end
                                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
                                Relative2[i,j,k]^Rpthroati:=abs(r);
                                goto back5;
                            end
                            else goto back6;
                        end
                    end
                back6:
                if (i=ArrSize+1) then
                    begin
                        if (Relative2[i,j,k]^Rpthroati>Relative[i-1,j,k]^Rpbody) then
                            begin
                                u:=random(10000)/10000.1;
                                r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));

```

```

if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroati:=abs(r);
    goto back6;

                                end
                                else goto back7;

    end
    else goto back7;
back7:
if ((i>1) and (i<ArrSize+1)) then
    begin
        if      ((Relative2[i,j,k]^Rpthroati>Relative[i-1,j,k]^Rpbody)      or
(Relative2[i,j,k]^Rpthroati>Relative[i,j,k]^Rpbody)) then
            begin
                u:=random(10000)/10000.1;
r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroati:=abs(r);
    goto back7;

                                end
                                else goto back8;

    end
    else goto back8;
back8:
end;
writeln('Step#2.1 - Pore-throat size distribution in i direction: ok');
{j*****}
iter:=0;
For k:=1 to ArrSize Do
    For i:=1 to ArrSize Do
        For j:=1 to ArrSize+1 Do
            Begin
                back17:
                u:=random(10000)/10000.1;
r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;

```

```

        end
        else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
iter:=iter+1;
if (iter>1000000) then goto back18;
if (r<rpthroatmin) then goto back17;
back18:
ii:=random(ArrSize);
jj:=random(ArrSize+1);
kk:=random(ArrSize);
if (Relative2[ii,jj,kk]^Rpthroatj=0) then
    begin
        Relative2[ii,jj,kk]^Rpthroatj:=abs(r);
    end
    else
    begin
        goto back18;
    end;
end;
j:=ArrSize+1;
For i:=1 to ArrSize Do
    For k:=1 to ArrSize Do
        Begin
            u:=random(10000)/10000.1;
            r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
            if (r<rpthroatmin) then
                begin
                    r:=r+10*rpthroatstddev;
                end
                else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
            Relative2[i,j,k]^Rpthroatj:=abs(r);
        end;
    i:=ArrSize;
    For j:=1 to ArrSize+1 Do
        For k:=1 to ArrSize Do
            Begin
                u:=random(10000)/10000.1;
                r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                if (r<rpthroatmin) then
                    begin
                        r:=r+10*rpthroatstddev;
                    end
                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
                Relative2[i,j,k]^Rpthroatj:=abs(r);
            end;

```

```

k:=ArrSize;
For j:=1 to ArrSize+1 Do
  For i:=1 to ArrSize Do
    Begin
      u:=random(10000)/10000.1;
      r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
      if (r<rpthroatmin) then
        begin
          r:=r+10*rpthroatstddev;
        end
      else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
      Relative2[i,j,k]^Rpthroatj:=abs(r);
    end;
  j:=0;
  For i:=1 to ArrSize Do
    For k:=1 to ArrSize Do
      Begin
        Relative2[i,j,k]^Rpthroatj:=0;
      end;
    For i:=1 to ArrSize Do
      For j:=1 to ArrSize+1 Do
        For k:=1 to ArrSize Do
          Begin
            back9:
            if (j=1) then
              begin
                if (Relative2[i,j,k]^Rpthroatj>Relative[i,j,k]^Rpbody) then
                  begin
                    u:=random(10000)/10000.1;
                    r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                    if (r<rpthroatmin) then
                      begin
                        r:=r+10*rpthroatstddev;
                      end
                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
                    Relative2[i,j,k]^Rpthroatj:=abs(r);
                    goto back9;
                  end
                else goto back10;
              end
            else goto back10;
          end
        back10:
        if (j=ArrSize+1) then
          begin
            if (Relative2[i,j,k]^Rpthroatj>Relative[i,j-1,k]^Rpbody) then

```

```

begin
    u:=random(10000)/10000.1;
    r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
    if (r<rpthroatmin) then
        begin
            r:=r+10*rpthroatstddev;
        end
        else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
    Relative2[i,j,k]^Rpthroatj:=abs(r);
        goto back10;
    end
    else goto back11;
end
else goto back11;
back11:
if ((j>1) and (j<ArrSize+1)) then
    begin
        if ((Relative2[i,j,k]^Rpthroatj>Relative[i,j-1,k]^Rpbody) or
(Relative2[i,j,k]^Rpthroatj>Relative[i,j,k]^Rpbody)) then
            begin
                u:=random(10000)/10000.1;
                r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                if (r<rpthroatmin) then
                    begin
                        r:=r+10*rpthroatstddev;
                    end
                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
                Relative2[i,j,k]^Rpthroatj:=abs(r);
                    goto back11;
                end
                else goto back12;
            end
            else goto back12;
        back12:
    end;
writeln('Step#2.2 - Pore-throat size distribution in j direction: ok');
{k*****}
iter:=0;
For i:=1 to ArrSize Do
    For j:=1 to ArrSize Do
        For k:=1 to ArrSize+1 Do
            Begin
                back19:
                u:=random(10000)/10000.1;

```



```

r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
if (r<rpthroatmin) then
    begin
        r:=r+10*rpthroatstddev;
    end
    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
iter:=iter+1;
if (iter>1000000) then goto back20;
if (r<rpthroatmin) then goto back19;
back20:
ii:=random(ArrSize);
jj:=random(ArrSize);
kk:=random(ArrSize+1);
if (Relative2[ii,jj,kk]^Rpthroatk=0) then
    begin
        Relative2[ii,jj,kk]^Rpthroatk:=abs(r);
    end
    else
    begin
        goto back20;
    end;
end;
k:=ArrSize+1;
For i:=1 to ArrSize Do
    For j:=1 to ArrSize Do
        Begin
            u:=random(10000)/10000.1;
            r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
            if (r<rpthroatmin) then
                begin
                    r:=r+10*rpthroatstddev;
                end
                else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
            Relative2[i,j,k]^Rpthroatk:=abs(r);
        end;
    i:=ArrSize;
    For j:=1 to ArrSize Do
        For k:=1 to ArrSize+1 Do
            Begin
                u:=random(10000)/10000.1;
                r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                if (r<rpthroatmin) then
                    begin
                        r:=r+10*rpthroatstddev;
                    end

```

```

        else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroatk:=abs(r);
end;
j:=ArrSize;
For i:=1 to ArrSize Do
  For k:=1 to ArrSize+1 Do
    Begin
      u:=random(10000)/10000.1;
      r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
      if (r<rpthroatmin) then
        begin
          r:=r+10*rpthroatstddev;
        end
        else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroatk:=abs(r);
end;
k:=0;
For i:=1 to ArrSize Do
  For j:=1 to ArrSize Do
    Begin
      Relative2[i,j,k]^Rpthroatk:=0;
    end;
  For i:=1 to ArrSize Do
    For j:=1 to ArrSize Do
      For k:=1 to ArrSize+1 Do
        Begin
          back13:
          if (k=1) then
            begin
              if (Relative2[i,j,k]^Rpthroatk>Relative[i,j,k]^Rpbody) then
                begin
                  u:=random(10000)/10000.1;
                  r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
                  if (r<rpthroatmin) then
                    begin
                      r:=r+10*rpthroatstddev;
                    end
                    else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
Relative2[i,j,k]^Rpthroatk:=abs(r);
                    goto back13;
                end
                else goto back14;
            end
            else goto back14;
          end
        end
      end
    end
  end
  back14:

```

```

if (k=ArrSize+1) then
  begin
    if (Relative2[i,j,k]^Rpthroatk>Relative[i,j,k-1]^Rpbody) then
      begin
        u:=random(10000)/10000.1;
        r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
        if (r<rpthroatmin) then
          begin
            r:=r+10*rpthroatstddev;
            end
          else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
        Relative2[i,j,k]^Rpthroatk:=abs(r);
        goto back14;
      end
    else goto back15;
  end
back15:
if ((k>1) and (k<ArrSize+1)) then
  begin
    if ((Relative2[i,j,k]^Rpthroatk>Relative[i,j,k-1]^Rpbody) or
(Relative2[i,j,k]^Rpthroatk>Relative[i,j,k]^Rpbody)) then
      begin
        u:=random(10000)/10000.1;
        r:=rpthroatmin*(power(-Ln(u),1/rpthroatmax));
        if (r<rpthroatmin) then
          begin
            r:=r+10*rpthroatstddev;
            end
          else if (r>rpthroatmax) then r:=r-10*rpthroatstddev;
        Relative2[i,j,k]^Rpthroatk:=abs(r);
        goto back15;
      end
    else goto back16;
  end
back16:
end;
writeln('Step#2.3 - Pore-throat size distribution in k direction: ok');
For ii:=1 to ArrSize+1 Do
  For jj:=1 to ArrSize Do
    For kk:=1 to ArrSize Do
      Begin
        writeln(Rptsizei,ii,' ',jj,' ',kk,',',Relative2[ii,jj,kk]^Rpthroati);
      end;

```

```

For ii:=1 to ArrSize Do
  For jj:=1 to ArrSize+1 Do
    For kk:=1 to ArrSize Do
      Begin
        writeln(Rptsizej,ii,' ',jj,' ',kk,",Relative2[ii,jj,kk]^Rpthroatj);
      end;
    For ii:=1 to ArrSize Do
      For jj:=1 to ArrSize Do
        For kk:=1 to ArrSize+1 Do
          Begin
            writeln(Rptsizek,ii,' ',jj,' ',kk,",Relative2[ii,jj,kk]^Rpthroatk);
          end;
        Close(Rptsizek);
      Close(Rptsizej);
    Close(Rptsizei);
  Close(Rpbsize);
end;
{ ***** }
Procedure Conductances;
var rr : Real;
    Gpbnw,Gpbw,Gptinw,Gptiw,Gptjnw,Gptjw,Gptknw,Gptkw,Gpb1f : text;
Begin
  Assign(Gpbnw,'C:\Gpbnw.txt');
  Rewrite(Gpbnw);
  Assign(Gpbw,'C:\Gpbw.txt');
  Rewrite(Gpbw);
  Assign(Gptinw,'C:\Gptinw.txt');
  Rewrite(Gptinw);
  Assign(Gptiw,'C:\Gptiw.txt');
  Rewrite(Gptiw);
  Assign(Gptjnw,'C:\Gptjnw.txt');
  Rewrite(Gptjnw);
  Assign(Gptjw,'C:\Gptjw.txt');
  Rewrite(Gptjw);
  Assign(Gptknw,'C:\Gptknw.txt');
  Rewrite(Gptknw);
  Assign(Gptkw,'C:\Gptkw.txt');
  Rewrite(Gptkw);
  Assign(Gpb1f,'C:\Gpb1f.txt');
  Rewrite(Gpb1f);
{ *****Non-wetting phase conductances in pore-bodies***** }
For k:=1 to ArrSize Do
  For j:=1 to ArrSize Do
    For i:=1 to ArrSize Do
      Begin

```

```

rr:=Relative[i,j,k]^Rpbody;
Relative[i,j,k]^Gpbodynw:=Gnw(rr,2*rr);
end;
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize Do
Begin
writeln(Gpbnw,i,'j','k',Relative[i,j,k]^Gpbodynw);
end;
{*****Wetting phase conductances in pore-bodies*****}
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize Do
Begin
rr:=Relative[i,j,k]^Rpbody;
Relative[i,j,k]^Gpbodyw:=Gw(rr,2*rr);
end;
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize Do
Begin
writeln(Gpbw,i,'j','k',Relative[i,j,k]^Gpbodyw);
end;
writeln('Step#3.1 - Pore-body conductances: ok');
{*****Non-wetting phase conductances in i-axis pore-throats*****}
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize+1 Do
Begin
rr:=Relative2[i,j,k]^Rpthroati;
Relative2[i,j,k]^Gpthroatinw:=Gnw(rr,2*rr);
end;
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize+1 Do
Begin
writeln(Gptinw,i,'j','k',Relative2[i,j,k]^Gpthroatinw);
end;
{*****Wetting phase conductances in i-axis pore-throats*****}
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
For i:=1 to ArrSize+1 Do
Begin
rr:=Relative2[i,j,k]^Rpthroati;
Relative2[i,j,k]^Gpthroatiw:=Gw(rr,2*rr);

```

```

        end;
    For k:=1 to ArrSize Do
    For j:=1 to ArrSize Do
        For i:=1 to ArrSize+1 Do
            Begin
                writeln(Gptiw,i,'j','k','Relative2[i,j,k]^Gpthroatiw);
            end;
        writeln('Step#3.2 - Conductances in i-axis Pore-throats: ok');
        {****Non-wetting phase conductances in j-axis pore-throats*****}
    For k:=1 to ArrSize Do
    For i:=1 to ArrSize Do
        For j:=1 to ArrSize+1 Do
            Begin
                rr:=Relative2[i,j,k]^Rpthroatj;
                Relative2[i,j,k]^Gpthroatjnw:=Gnw(rr,2*rr);
            end;
        For k:=1 to ArrSize Do
        For i:=1 to ArrSize Do
            For j:=1 to ArrSize+1 Do
                Begin
                    writeln(Gptjnw,i,'j','k','Relative2[i,j,k]^Gpthroatjnw);
                end;
            {*****Wetting phase conductances in j-axis pore-throats*****}
        For k:=1 to ArrSize Do
        For i:=1 to ArrSize Do
            For j:=1 to ArrSize+1 Do
                Begin
                    rr:=Relative2[i,j,k]^Rpthroatj;
                    Relative2[i,j,k]^Gpthroatjw:=Gw(rr,2*rr);
                end;
            For k:=1 to ArrSize Do
            For i:=1 to ArrSize Do
                For j:=1 to ArrSize+1 Do
                    Begin
                        writeln(Gptjw,i,'j','k','Relative2[i,j,k]^Gpthroatjw);
                    end;
                writeln('Step#3.3 - Conductances in j-axis Pore-throats: ok');
                {****Non-wetting phase conductances in k-axis pore-throats*****}
            For i:=1 to ArrSize Do
            For j:=1 to ArrSize Do
                For k:=1 to ArrSize+1 Do
                    Begin
                        rr:=Relative2[i,j,k]^Rpthroatk;
                        Relative2[i,j,k]^Gpthroatknw:=Gnw(rr,2*rr);
                    end;

```

```

For i:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For k:=1 to ArrSize+1 Do
    Begin
      writeln(Gptknw,i,'j','k',' ',Relative2[i,j,k]^Gpthroatknw);
    end;
  {*****Wetting phase conductances in k-axis pore-throats*****}
For i:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For k:=1 to ArrSize+1 Do
    Begin
      rr:=Relative2[i,j,k]^Rpthroatk;
      Relative2[i,j,k]^Gpthroatkw:=Gw(rr,2*rr);
    end;
For i:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For k:=1 to ArrSize+1 Do
    Begin
      writeln(Gptkw,i,'j','k',' ',Relative2[i,j,k]^Gpthroatkw);
    end;
writeln('Step#3.4 - Conductances in k-axis Pore-throats: ok');
{*****1-phase flow conductances in pores*****}
For i:=1 to ArrSize Do
  For j:=1 to ArrSize Do
    For k:=1 to ArrSize Do
      Begin
        Relative[i,j,k]^Gpb1fnw:=Relative[i,j,k]^Gpbodynw+Relative[i,j,k]^Gpbod
        yw*MuW/MuO;
        Relative[i,j,k]^Gpb1fw:=Relative[i,j,k]^Gpbodyw+Relative[i,j,k]^Gpbod
        yw*MuO/MuW;
        writeln(Gpb1f,i,'j','k',' ',Relative[i,j,k]^Gpb1fnw,'
        ',Relative[i,j,k]^Gpb1fw);
      end;
    {*****1-phase flow conductances in i-axis pore-throats*****}
  For k:=1 to ArrSize Do
    For j:=1 to ArrSize Do
      For i:=1 to ArrSize+1 Do
        Begin
          Relative2[i,j,k]^Gpti1fnw:=Relative2[i,j,k]^Gpthroatinw+Relative2[i,j,k]^
          Gpthroatiw*MuW/MuO;
          Relative2[i,j,k]^Gpti1fw:=Relative2[i,j,k]^Gpthroatiw+Relative2[i,j,k]^G
          pthroatinw*MuO/MuW;
        end;

```

```

{*****1-phase flow conductances in j-axis pore-throats*****}
For k:=1 to ArrSize Do
For i:=1 to ArrSize Do
  For j:=1 to ArrSize+1 Do
    Begin
      Relative2[i,j,k]^Gptj1fnw:=Relative2[i,j,k]^Gpthroatjnw+Relative2[i,j,k]^
      Gpthroatjw*MuW/MuO;
      Relative2[i,j,k]^Gptj1fw:=Relative2[i,j,k]^Gpthroatjw+Relative2[i,j,k]^G
      pthroatjnw*MuO/MuW;
    end;
  {*****1-phase flow conductances in k-axis pore-throats*****}
  For i:=1 to ArrSize Do
  For j:=1 to ArrSize Do
    For k:=1 to ArrSize+1 Do
      Begin
        Relative2[i,j,k]^Gptk1fnw:=Relative2[i,j,k]^Gpthroatknw+Relative2[i,j,k]
        ^Gpthroatkw*MuW/MuO;
        Relative2[i,j,k]^Gptk1fw:=Relative2[i,j,k]^Gpthroatkw+Relative2[i,j,k]^
        Gpthroatknw*MuO/MuW;
      end;
    Close(Gpb1f);
    Close(Gptkw);
    Close(Gptknw);
    Close(Gptjw);
    Close(Gptjnw);
    Close(Gptiw);
    Close(Gptinw);
    Close(Gpbw);
    Close(Gpbnw);
  End;
  Procedure CapillaryPressures;
  var rr : Real;
  Pcpb,Pcpti,Pcptj,Pcptk : text;
  Begin
    Assign(Pcpb,'C:\Pcpb.txt');
    Rewrite(Pcpb);
    Assign(Pcpti,'C:\Pcpti.txt');
    Rewrite(Pcpti);
    Assign(Pcptj,'C:\Pcptj.txt');
    Rewrite(Pcptj);
    Assign(Pcptk,'C:\Pcptk.txt');
    Rewrite(Pcptk);
  {*****Capillary Pressure in Pore-bodies*****}
  For k:=1 to ArrSize Do
  For j:=1 to ArrSize Do

```



```

For i:=1 to ArrSize Do
  Begin
    rr:=Relative[i,j,k]^Rpbody;
    Relative[i,j,k]^Pcpbody:=Pcpbpt(rr);
    Relative[i,j,k]^Pctemp:=Relative[i,j,k]^Pcpbody;
  end;
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For i:=1 to ArrSize Do
    Begin
      writeln(Pcpb,i,'j','k',Relative[i,j,k]^Pcpbody);
    end;
writeln('Step#4.1 - Capillary Pressures in Pore-bodies: ok');
{ *****Capillary pressure in i-axis throats***** }
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For i:=1 to ArrSize+1 Do
    Begin
      rr:=Relative2[i,j,k]^Rpthroati;
      Relative2[i,j,k]^Pcpthroati:=Pcpbpt(rr);
    end;
For k:=1 to ArrSize Do
For j:=1 to ArrSize Do
  For i:=1 to ArrSize+1 Do
    Begin
      writeln(Pcpti,i,'j','k',Relative2[i,j,k]^Pcpthroati);
    end;
writeln('Step#4.2 - Capillary Pressures in i-axis Pore-throats: ok');
{ *****Capillary pressure in j-axis throats***** }
For k:=1 to ArrSize Do
For i:=1 to ArrSize Do
  For j:=1 to ArrSize+1 Do
    Begin
      rr:=Relative2[i,j,k]^Rpthroatj;
      Relative2[i,j,k]^Pcpthroatj:=Pcpbpt(rr);
    end;
For k:=1 to ArrSize Do
For i:=1 to ArrSize Do
  For j:=1 to ArrSize+1 Do
    Begin
      writeln(Pcptj,i,'j','k',Relative2[i,j,k]^Pcpthroatj);
    end;
writeln('Step#4.3 - Capillary Pressures in j-axis Pore-throats: ok');
{ *****Capillary pressure in k-axis throats***** }
For i:=1 to ArrSize Do

```

```

For j:=1 to ArrSize Do
  For k:=1 to ArrSize+1 Do
    Begin
      rr:=Relative2[i,j,k]^Rpthroatk;
      Relative2[i,j,k]^Pcpthroatk:=Pcpbpt(rr);
    end;
  For i:=1 to ArrSize Do
    For j:=1 to ArrSize Do
      For k:=1 to ArrSize+1 Do
        Begin
          writeln(Pcptk,i,' ',j,' ',k,' ',Relative2[i,j,k]^Pcpthroatk);
        end;
      writeln('Step#4.4 - Capillary Pressures in k-axis Pore-throats: ok');
      close(Pcptk);
      close(Pcptj);
      close(Pcpti);
      close(Pcpb);
    End;
  Procedure Conductances_btw_pores;
  var Conpbi,Conpbj,Conpbk: text;
  begin
    Assign(Conpbi,'C:\Conpbi.txt');
    Rewrite(Conpbi);
    For k:=1 to ArrSize Do
      For j:=1 to ArrSize Do
        For i:=1 to ArrSize Do
          Begin
            Relative[i,j,k]^Giinw:=2*Relative[i,j,k]^Gpbodynw*Relative[i+1,j,k]^Gp
            bodynw*Relative2[i+1,j,k]^Gpthroatinw/
            (2*Relative[i,j,k]^Gpbodynw*Relative[i+1,j,k]^Gpbodynw+Relative2[i+1,
            j,k]^Gpthroatinw*(Relative[i,j,k]^Gpbodynw+Relative[i+1,j,k]^Gpbodyn
            w));
            Relative[i,j,k]^Giiw:=2*Relative[i,j,k]^Gpbodyw*Relative[i+1,j,k]^Gpbo
            dyw*Relative2[i+1,j,k]^Gpthroatiw/
            (2*Relative[i,j,k]^Gpbodyw*Relative[i+1,j,k]^Gpbodyw+Relative2[i+1,j,k
            ]^Gpthroatiw*(Relative[i,j,k]^Gpbodyw+Relative[i+1,j,k]^Gpbodyw));
            Relative[i,j,k]^Gii1fnw:=2*Relative[i,j,k]^Gpb1fnw*Relative[i+1,j,k]^Gp
            b1fnw*Relative2[i+1,j,k]^Gpti1fnw/
            (2*Relative[i,j,k]^Gpb1fnw*Relative[i+1,j,k]^Gpb1fnw+Relative2[i+1,j,k]
            ^Gpti1fnw*(Relative[i,j,k]^Gpb1fnw+Relative[i+1,j,k]^Gpb1fnw));
            Relative[i,j,k]^Gii1fw:=2*Relative[i,j,k]^Gpb1fw*Relative[i+1,j,k]^Gpb1
            fw*Relative2[i+1,j,k]^Gpti1fw/
            (2*Relative[i,j,k]^Gpb1fw*Relative[i+1,j,k]^Gpb1fw+Relative2[i+1,j,k]^
            Gpti1fw*(Relative[i,j,k]^Gpb1fw+Relative[i+1,j,k]^Gpb1fw));

```

```

writeln(Conpbi,i,'j','k','Relative[i,j,k]^Giinw','Relative[i,j,k]^Giiw','
    Relative[i,j,k]^Gii1fnw','Relative[i,j,k]^Gii1fw);
End;
writeln('Step#5.1 - Conductances between pores in i-direction: ok');
Close(Conpbi);
Assign(Conpbj,'C:\Conpbj.txt');
Rewrite(Conpbj);
  For i:=1 to ArrSize Do
    For k:=1 to ArrSize Do
      For j:=1 to ArrSize-1 Do
        Begin
          Relative[i,j,k]^Gjjnw:=2*Relative[i,j,k]^Gpbodyn*Relative[i,j+1,k]^Gp
            bodynw*Relative2[i,j+1,k]^Gpthroatjnw/
            (2*Relative[i,j,k]^Gpbodyn*Relative[i,j+1,k]^Gpbodyn+Relative2[i,j+
              1,k]^Gpthroatjnw*(Relative[i,j,k]^Gpbodyn+Relative[i,j+1,k]^Gpbodyn
                w));
          Relative[i,j,k]^Gjjw:=2*Relative[i,j,k]^Gpbodyw*Relative[i,j+1,k]^Gpbo
            dyw*Relative2[i,j+1,k]^Gpthroatjw/
            (2*Relative[i,j,k]^Gpbodyw*Relative[i,j+1,k]^Gpbodyw+Relative2[i,j+1,k
              ]^Gpthroatjw*(Relative[i,j,k]^Gpbodyw+Relative[i,j+1,k]^Gpbodyw));
          Relative[i,j,k]^Gjj1fnw:=2*Relative[i,j,k]^Gpb1fnw*Relative[i,j+1,k]^Gp
            b1fnw*Relative2[i,j+1,k]^Gptj1fnw/
            (2*Relative[i,j,k]^Gpb1fnw*Relative[i,j+1,k]^Gpb1fnw+Relative2[i,j+1,k]
              ^Gptj1fnw*(Relative[i,j,k]^Gpb1fnw+Relative[i,j+1,k]^Gpb1fnw));
          Relative[i,j,k]^Gjj1fw:=2*Relative[i,j,k]^Gpb1fw*Relative[i,j+1,k]^Gpb1
            fw*Relative2[i,j+1,k]^Gptj1fw/
            (2*Relative[i,j,k]^Gpb1fw*Relative[i,j+1,k]^Gpb1fw+Relative2[i,j+1,k]^
              Gptj1fw*(Relative[i,j,k]^Gpb1fw+Relative[i,j+1,k]^Gpb1fw));
          writeln(Conpbj,i,'j','k','Relative[i,j,k]^Gjjnw','Relative[i,j,k]^Gjjw','
            Relative[i,j,k]^Gjj1fnw','Relative[i,j,k]^Gjj1fw);
        End;
      writeln('Step#5.2 - Conductances between pores in j-direction: ok');
    Close(Conpbj);
    Assign(Conpbk,'C:\Conpbk.txt');
    Rewrite(Conpbk);
      For j:=1 to ArrSize Do
        For i:=1 to ArrSize Do
          For k:=1 to ArrSize-1 Do
            Begin
              Relative[i,j,k]^Gkknw:=2*Relative[i,j,k]^Gpbodyn*Relative[i,j,k+1]^G
                pbodyn*Relative2[i,j,k+1]^Gpthroatknw/
                (2*Relative[i,j,k]^Gpbodyn*Relative[i,j,k+1]^Gpbodyn+Relative2[i,j,k
                  +1]^Gpthroatknw*
                (Relative[i,j,k]^Gpbodyn+Relative[i,j,k+1]^Gpbodyn));
            End;
          End;
        End;
      End;
    End;
  End;

```

```

Relative[i,j,k]^Gkkw:=2*Relative[i,j,k]^Gpbodyw*Relative[i,j,k+1]^Gpb
odyw*Relative2[i,j,k+1]^Gpthroatkw/
(2*Relative[i,j,k]^Gpbodyw*Relative[i,j,k+1]^Gpbodyw+Relative2[i,j,k+1]
]^Gpthroatkw*(Relative[i,j,k]^Gpbodyw+Relative[i,j,k+1]^Gpbodyw));
Relative[i,j,k]^Gkk1fnw:=2*Relative[i,j,k]^Gpb1fnw*Relative[i,j,k+1]^G
pb1fnw*Relative2[i,j,k+1]^Gptk1fnw/
(2*Relative[i,j,k]^Gpb1fnw*Relative[i,j,k+1]^Gpb1fnw+Relative2[i,j,k+1]
]^Gptk1fnw*(Relative[i,j,k]^Gpb1fnw+Relative[i,j,k+1]^Gpb1fnw));
Relative[i,j,k]^Gkk1fw:=2*Relative[i,j,k]^Gpb1fw*Relative[i,j,k+1]^Gpb
1fw*Relative2[i,j,k+1]^Gptk1fw/
(2*Relative[i,j,k]^Gpb1fw*Relative[i,j,k+1]^Gpb1fw+Relative2[i,j,k+1]^
Gptk1fw*(Relative[i,j,k]^Gpb1fw+Relative[i,j,k+1]^Gpb1fw));
writeln(Conpbk,i,' 'j,' 'k,' ',Relative[i,j,k]^Gkknw,' ',Relative[i,j,k]^Gkkw,'
',Relative[i,j,k]^Gkk1fnw,' ',Relative[i,j,k]^Gkk1fw);
    End;
writeln('Step#5.3 - Conductances between pores in k-direction: ok');
Close(Conpbk);
end;
Procedure Drainage;
var A,B,C,D,E,F,TOT,Pijk,Pdif,
    AA,BB,CC,DD,EE,FF,TOTnw,Pijknw,Pdifnw,
    AAA,BBB,CCC,DDD,EEE,FFF,Totw,Pijkw,Pdifw,
    Qnwtot,Qwtot,Q1fnwtot,Q1fwtot,voltot,voltot2,Sw,
    Gii1fnwtot,Giinwtot,Gii1fwtot,Gii1fwtot2,temporary,
    krwq,krnwq,x1,x2,y1,y2,c1,c2,Pcpbdy,krwet,krownwet,
    Swfwp:real;
ijk:longint;
iii,jjj,kkk:integer;
pressure,kr,Qu,krQu,krdr:text;
Begin
Assign(pressure,'C:\Press.txt');
Rewrite(pressure);
Assign(kr,'C:\kr.txt');
Rewrite(kr);
Assign(Qu,'C:\Qu.txt');
Rewrite(Qu);
Assign(krQu,'C:\krQu.txt');
Rewrite(krQu);
ijk:=0;
Repeat
For i:=1 to ArrSize-1 Do
For j:=1 to ArrSize-1 Do
For k:=1 to ArrSize-1 Do
Begin

```

$A := \text{Relative}[i-1, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i-1, j, k]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P - \text{Relative2}[i-1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i-1, j, k]^{\wedge} . \text{Giiw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i-1, j, k]^{\wedge} . P + \text{Relative}[i-1, j, k]^{\wedge} . \text{Pcpbody});$
 $B := \text{Relative}[i, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i+1, j, k]^{\wedge} . P - \text{Relative2}[i+1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Giiw} * H(\text{Relative}[i+1, j, k]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$
 $C := \text{Relative}[i, j, k]^{\wedge} . \text{Gjjnw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i, j-1, k]^{\wedge} . P - \text{Relative2}[i, j-1, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Gjjw} * H(\text{Relative}[i, j-1, k]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$
 $D := \text{Relative}[i, j, k]^{\wedge} . \text{Gjjnw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i, j+1, k]^{\wedge} . P - \text{Relative2}[i, j+1, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Gjjw} * H(\text{Relative}[i, j+1, k]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$
 $E := \text{Relative}[i, j, k]^{\wedge} . \text{Gkknw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i, j, k-1]^{\wedge} . P - \text{Relative2}[i, j, k-1]^{\wedge} . \text{Pcpthroatk})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Gkkw} * H(\text{Relative}[i, j, k-1]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$
 $F := \text{Relative}[i, j, k]^{\wedge} . \text{Gkknw} * H(\text{Relative}[i, j, k]^{\wedge} . P - \text{Relative}[i, j, k+1]^{\wedge} . P - \text{Relative2}[i, j, k+1]^{\wedge} . \text{Pcpthroatk})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Gkkw} * H(\text{Relative}[i, j, k+1]^{\wedge} . P - \text{Relative}[i, j, k]^{\wedge} . P + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$
 $\text{TOT} := A + B + C + D + E + F;$
 $\text{Pijk} := (A * (\text{Relative}[i-1, j, k]^{\wedge} . P - \text{Relative2}[i-1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + B * (\text{Relative}[i+1, j, k]^{\wedge} . P + \text{Relative2}[i+1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + C * (\text{Relative}[i, j-1, k]^{\wedge} . P + \text{Relative2}[i, j-1, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + D * (\text{Relative}[i, j+1, k]^{\wedge} . P + \text{Relative2}[i, j+1, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + E * (\text{Relative}[i, j, k-1]^{\wedge} . P + \text{Relative2}[i, j, k-1]^{\wedge} . \text{Pcpthroatk})$
 $\quad + F * (\text{Relative}[i, j, k+1]^{\wedge} . P + \text{Relative2}[i, j, k+1]^{\wedge} . \text{Pcpthroatk})) / \text{TOT};$
 $\text{Pdif} := \text{abs}(\text{Pijk} - \text{Relative}[i, j, k]^{\wedge} . P);$
if (Pdif > error) and (Pijk > 0) then $\text{Relative}[i, j, k]^{\wedge} . P := \text{Pijk};$
 $\text{AA} := \text{Relative}[i-1, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i-1, j, k]^{\wedge} . \text{Pnw} - \text{Relative}[i, j, k]^{\wedge} . \text{Pnw} - \text{Relative2}[i-1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i-1, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i, j, k]^{\wedge} . \text{Pnw} - \text{Relative}[i-1, j, k]^{\wedge} . \text{Pnw} + \text{Relative}[i-1, j, k]^{\wedge} . \text{Pcpbody});$
 $\text{AAA} := \text{Relative}[i-1, j, k]^{\wedge} . \text{Giiw} * H(\text{Relative}[i-1, j, k]^{\wedge} . \text{Pw} - \text{Relative}[i, j, k]^{\wedge} . \text{Pw} - \text{Relative2}[i-1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i-1, j, k]^{\wedge} . \text{Giiw} * H(\text{Relative}[i, j, k]^{\wedge} . \text{Pw} - \text{Relative}[i-1, j, k]^{\wedge} . \text{Pw} + \text{Relative}[i-1, j, k]^{\wedge} . \text{Pcpbody});$
 $\text{BB} := \text{Relative}[i, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i, j, k]^{\wedge} . \text{Pnw} - \text{Relative}[i+1, j, k]^{\wedge} . \text{Pnw} - \text{Relative2}[i+1, j, k]^{\wedge} . \text{Pcpthroati})$
 $\quad + \text{Relative}[i, j, k]^{\wedge} . \text{Giinw} * H(\text{Relative}[i+1, j, k]^{\wedge} . \text{Pnw} - \text{Relative}[i, j, k]^{\wedge} . \text{Pnw} + \text{Relative}[i, j, k]^{\wedge} . \text{Pcpbody});$


```

Pijkw:=(AAA*(Relative[i-1,j,k]^Pw-Relative2[i-1,j,k]^Pcpthroati)
+BBB*(Relative[i+1,j,k]^Pw+Relative2[i+1,j,k]^Pcpthroati)
+CCC*(Relative[i,j-1,k]^Pw+Relative2[i,j-1,k]^Pcpthroatj)
+DDD*(Relative[i,j+1,k]^Pw+Relative2[i,j+1,k]^Pcpthroatj)
+EEE*(Relative[i,j,k-1]^Pw+Relative2[i,j,k-1]^Pcpthroatk)
+FFF*(Relative[i,j,k+1]^Pw+Relative2[i,j,k+1]^Pcpthroatk))/TOTw;
Pdifnw:=abs(Pijkw-Relative[i,j,k]^Pnw);
Pdifw:=abs(Pijkw-Relative[i,j,k]^Pw);
if (Pdifnw>error) and (Pijkw>0) then Relative[i,j,k]^Pnw:=Pijkw;
if (Pdifw>error) and (Pijkw>0) then Relative[i,j,k]^Pw:=Pijkw;
end;
ijk:=ijk+1;
until ijk=1000;
For i:=2 to ArrSize-1 Do
  For j:=2 to ArrSize-1 Do
    For k:=2 to ArrSize-1 Do
      Begin
        writeln(pressure,i,' ',j,' ',k,' ',Relative[i,j,k]^P,' ',Relative[i,j,k]^Pnw,'
',Relative[i,j,k]^Pw);
      end;
    Qnwtot:=0;
    Qwtot:=0;
    Q1fnwtot:=0;
    Q1fwtot:=0;
    Voltot:=0;
    Gii1fnwtot:=0;
    Giinwtot:=0;
    Gii1fwtot:=0;
    krwq:=0;
    krnwq:=0;
    For i:=2 to ArrSize-1 Do
      For j:=2 to ArrSize-1 Do
        For k:=2 to ArrSize-1 Do
          Begin
            voltot:=voltot+power(2*relative[i,j,k]^Rpbbody,3);
            Relative[i,j,k]^Qnw:=Relative[i,j,k]^Giinw*(Relative[i,j,k]^P-Relative[i-
            1,j,k]^P);
            Gii1fnwtot:=Gii1fnwtot+Relative[i,j,k]^Gii1fnw;
            Relative[i,j,k]^Qw:=Relative[i,j,k]^Giinw*(Relative[i,j,k]^P-Relative[i-
            1,j,k]^P);
            Gii1fwtot:=Gii1fwtot+Relative[i,j,k]^Gii1fw;
            Relative[i,j,k]^Q1fnw:=Relative[i,j,k]^Gii1fnw*(Relative[i,j,k]^Pnw-
            Relative[i-1,j,k]^Pnw);
            Q1fnwtot:=Q1fnwtot+Relative[i,j,k]^Q1fnw;

```

```

Relative[i,j,k]^Q1fw:=Relative[i,j,k]^Gii1fw*(Relative[i,j,k]^Pw-
Relative[i-1,j,k]^Pw);
Q1fwtot:=Q1fwtot+Relative[i,j,k]^Q1fw;
writeln(Qu,i,' ',j,' ',k,' ',Relative[i,j,k]^Qnw,' ',Relative[i,j,k]^Qw,' ',
Relative[i,j,k]^Q1fnw,' ',Relative[i,j,k]^Q1fw);
end;
Gii1fwtot2:=Gii1fwtot;
voltot2:=voltot;
For ii:=2 to ArrSize-1 do
For jj:=2 to ArrSize-1 Do
For kk:=2 to ArrSize-1 Do
begin
temporary:=1e+30;
For iii:=2 to ArrSize-1 Do
For jjj:=2 to ArrSize-1 Do
For kkk:=2 to ArrSize-1 Do
Begin
if temporary>relative[iii,jjj,kkk]^Pctemp then
begin
temporary:=relative[iii,jjj,kkk]^Pctemp;
i:=iii;
j:=jjj;
k:=kkk;
end;
end;
voltot2:=voltot2-4/3*Pi*power(relative[i,j,k]^Rpbbody,3);
Sw:=voltot2/voltot;
Gii1fwtot2:=Gii1fwtot2-(Relative[i,j,k]^Gii1fw{-
Relative[i,j,k]^Giinw});
Relative[i,j,k]^krw:=Gii1fwtot2/Gii1fwtot;
if (Relative[i,j,k]^krw<0) then Relative[i,j,k]^krw:=0;
Giinwtot:=Giinwtot+Relative[i,j,k]^Giinw;
Qnwtot:=Qnwtot+Relative[i,j,k]^Qnw;
Relative[i,j,k]^krnw:=Giinwtot/Gii1fnwtot;
relative[i,j,k]^Pctemp:=1e+30;
writeln(kr,i,' ',j,' ',k,' ',relative[i,j,k]^Pcpbody,' ',Sw,' ',Relative[i,j,k]^krw,'
',Relative[i,j,k]^krnw);

{ krwq:=krwq+Qnwtot/Q1fwtot;}
{y krnwq:=krnwq+Qnwtot/Q1fnwtot;}
writeln(krQu,i,' ',j,' ',k,' ',relative[i,j,k]^Pcpbody,' ',Sw,' ',krwq,' ',krnwq);
end;
x2:=1;
y2:=1;
x1:=Sw;

```



```

y1:=Relative[i,j,k]^.krw;
b:=1;
a:=(y1-y2)/(x1-x2);
c1:=(y1-y2)/(x1-x2)*x1-y1;
close(kr);
Assign(krdr,'C:\krdr.txt');
Rewrite(krdr);
Assign(kr,'C:\kr.txt');
Reset(kr);
writeln(krdr,'i',' ','j',' ','k',' ','Pc',' ','1.00E+00',' ','1.00E+00',' ','1.00E+00','
','0.00E+00');
repeat
  readln(kr,i,j,k,Pcpbody,Sw,krwet,knowet);
  c2:=abs(a*Sw+b*krwet+c1)+c1;
  Swfwp:=(b*b*Sw-a*b*krwet-a*c2)/(a*a+b*b);
  Relative[i,j,k]^.krw:=(a*a*krwet-a*b*Sw-b*c2)/(a*a+b*b);
  if (Relative[i,j,k]^.krw>1.0) then Relative[i,j,k]^.krw:=1.0;
  if (Relative[i,j,k]^.krw<0.0) then Relative[i,j,k]^.krw:=0.0;
  if (Swfwp>1.0) then Swfwp:=1.0;
  if (Swfwp<0.0) then Swfwp:=0.0;
  writeln(krdr,i,' ','j',' ','k',' ','relative[i,j,k]^.' Pcpbody, Swfwp,relative[i,j,k]^
.krw,Sw,relative[i,j,k]^ .krnw);
until eof(kr);
close(kr);
close(krdr);
close(krQu);
close(Qu);
close(pressure);
End;
begin{ main prog}
randomize;
clrscr;
  case ncorners of
    3: alfa:=(ncorners-2)*Pi/ncorners/2;
    4: alfa:=(ncorners-2)*Pi/ncorners/2;
    5: alfa:=(ncorners-2)*Pi/ncorners/2;
    6: alfa:=(ncorners-2)*Pi/ncorners/2;
  end;
AllocateArray_1;
initialize_1;
PoreSizeDistribution;
Conductances;
CapillaryPressures; Conductances_btw_pores;
Drainage; DisposeArray_1;
End.

```

APPENDIX B

SOURCE CODE OF THE ARTIFICIAL NEURAL

NETWORK PROGRAM

```
program XOR;
const inpmx = 50; { input units' maximum number }
      hlmax = 2; { hidden layer maximum number }
      humax = 30;
      exlmax = 50; { ex-layer unit maximum number }
      gama = 1; { b - a interval length of sigmoid function }
      nu = 0; { - a, a is the minimum limit of sigmoid }
           { b-a=1 & a=0 ==> binary sigmoid
           b-a=2 & a=-1 ==> bipolar sigmoid }
      sig = 1; { define shape of lazy-s }
type vector1 = array[1..inpmx] of real; {input, output and desired vectors
type}
      weightvec = array[1..hlmax+1,1..humax,1..exlmax] of real;
           { weight 3-d array.. 1= weight's active layer number
           2= That active layer's hidden number
           3= The connected before neuron number }
var inpvec,desvec : vector1;
    w,wmin : weightvec;
    wbias,outvec : array[1..hlmax+1,1..humax] of real;
    hunum : array[0..hlmax+1] of integer;
    bias : array[1..hlmax+1] of real;
    inum,onum,hlum,i,k,j,
    method,inyc,funcnum1,funcnum2 : integer;
    lr,mu,errmin,
    deserr,err,errt: real;
    ans : char;
    inpn,outn : string;
    inpf,outf : text;
    times,counter : integer; {used in testing}
function f(i:integer;a:real):real;
begin
```

```

case i of
  1 : f:=a; {identity function}
  2 : f:=gama/(1+exp(-sig*a))-nu; {sigmoid function}
  3 : f:=(exp(a)-exp(-a))/(exp(a)+exp(-a)); {hyperbolic tangent function}
  4 : f:=a*a; {square function}
  5 : f:=sin(a);{sinus function}
  6 : f:=2/pi*arctan(a); {arc tangent function}
  7 : if a >= 0 then f:={log}ln(1+a) {loagritmic function}
      else f:=-{log}ln(1-a);
  8 : f:=exp(-a*a); {gaussian function}
end;
end;
function df(i:integer;a:real):real;
begin
  case i of
    1 : df:=1;
    2 : df:=sig/gama*(nu+f(2,a))*(gama-nu-f(2,a));
    3 : df:=(1+f(3,a))*(1-f(3,a));
    4 : df:=2*a;
    5 : df:=cos(a);
    6 : df:=2/pi/(1+a*a);
    7 : if a >= 0 then df:=1/(1+a)
        else df:=1/(1-a);
    8 : df:=-2*a*exp(-a*a);
  end;
end;
procedure train;
var wex,wtemp : weightvec;
    del       : array[1..hlmax+1,0..humax] of real;
    epoch      : real;
    ANS2       : CHAR;
procedure pararead;
begin
  write('Enter the learning rate ( 0 < à < 1 )--> ');
  readln(lr);
  writeln(' :20,'1 - Backpropagation ');
  writeln(' :20,'2 - Backpropagation with momentum');
  write('Select The Backpropagation Method : ');
  readln(method);
  case method of
    1 : begin
        end;
    2 : begin
        write('Enter the momentum term value ( 0 < æ < 1 )--> ');
        readln(mu);
        end;
  end;
end;

```

```

end;
write('Enter the Desired Error value (  $0 < \hat{\epsilon} < 1$  )--> ');
readln(deserr);
writeln('What kind of initialization Method do you want ? ');
writeln(' :20,'1- Random Initialization');
writeln(' :20,'2- Read from a file');
write('Your Choise ==> ');
readln(inyc);
end;
procedure ran_initw;
begin
    randomize;
    for k:=1 to hlnum+1 do
        for j:=1 to hunum[k] do
            begin
                { wbias[k,j]:=-0.5+random(101)/100.1;}
                wbias[k,j]:=-0.1+random(101)/1000.1;
                for i:=1 to hunum[k-1] do
                    { w[k,j,i]:=-0.5+random(100)/100.; }
                    w[k,j,i]:=-0.1+random(101)/1000.1;
                end;
                if method=2 then wtemp:=w;
            end;
        end;
    end;
    procedure read_weight_from_file;
    begin
        reset(outf);
        for k:=1 to hlnum+1 do
            begin
                for j:=1 to hunum[k] do
                    begin
                        read(outf,wbias[k,j]);
                        for i:=1 to hunum[k-1] do
                            read(outf,w[k,j,i]);
                        readln(outf);
                    end;
                readln(outf);
            end;
        end;
        close(outf);
    end;
    procedure dataread;
    begin
        for i:=1 to inum do
            begin
                read(inpf,inpvec[i]);
            end;
        for i:=1 to onum do

```

```

        read(inpf,desvec[i]);
    readln(inpf);
end;
procedure forwd;
begin
    for k:=1 to hlnum-1 do
        for j:=1 to hunum[k] do
            begin
                outvec[k,j]:=bias[k]*wbias[k,j];
                for i:=1 to hunum[k-1] do
                    if k=1
                    then outvec[k,j]:=outvec[k,j]+w[k,j,i]*f(funcnum1,inpvec[i])
                    else outvec[k,j]:=outvec[k,j]+w[k,j,i]*f(funcnum2,outvec[k-1,i]);
                end;
            end;
        end;
    end;
    procedure control;
    begin
        for j:=1 to onum do
            begin
                err:=(desvec[j]-f(funcnum2,outvec[hlnum+1,j]));
                errt:=errt+0.5*(err*err);
                del[hlnum+1,j]:=err*df(funcnum2,outvec[hlnum+1,j]);
            end;
        end;
        for k:=hlnum downto 1 do
            for i:=1 to hunum[k] do
                begin
                    del[k,i]:=0;
                    for j:=1 to hunum[k+1] do
                        del[k,i]:=del[k,i]+del[k+1,j]*w[k+1,j,i];
                        del[j,i]:=del[k,i]*df(funcnum2,outvec[j,i]);
                    end;
                end;
            end;
        end;
    end;
    procedure backprop;
    begin
        for k:=1 to hlnum+1 do
            for j:=1 to hunum[k] do
                begin
                    wbias[k,j]:=wbias[k,j]+lr*del[k,j]*bias[k];
                    for i:=1 to hunum[k-1] do
                        if k=1
                        then w[k,j,i]:=w[k,j,i]+lr*del[k,j]*f(funcnum1,inpvec[i])
                        else w[k,j,i]:=w[k,j,i]+lr*del[k,j]*f(funcnum2,outvec[k-1,i]);
                    end;
                end;
            end;
        end;
    end;
    procedure back_moment;
    begin

```

```

wex:=wtemp; {old weight values}
wtemp:=w; {temporary weight values}
for k:=1 to hlnum+1 do
  for j:=1 to hunum[k] do
    begin
      wbias[k,j]:=wbias[k,j]+lr*del[k,j]*bias[k];
      for i:=1 to hunum[k-1] do
        if k=1
          then w[k,j,i]:=w[k,j,i]+lr*del[k,j]*f(funcnum1,inpvec[i])+
            mu*(w[k,j,i]-wex[k,j,i])
          else w[k,j,i]:=w[k,j,i]+lr*del[k,j]*f(funcnum2,outvec[k-1,i])+
            mu*(w[k,j,i]-wex[k,j,i]);
        end;
      end;
    end;
  procedure epoch_err_write;
  begin
    epoch:=epoch+1;
    writeln(epoch,' ',errt)
  end;
  procedure wwwrite;
  begin
    rewrite(outf);
    for k:=1 to hlnum+1 do
      begin
        for j:=1 to hunum[k] do
          begin
            write(outf,wbias[k,j]:12:8,' ');
            for i:=1 to hunum[k-1] do
              write(outf,w[k,j,i]:12:8,' ');
            writeln(outf);
          end;
        writeln(outf);
      end;
    end;
    close(outf)
  end;
  begin
    assign(inpf,inpfn);
    ans2:=' ';
    errmin:=maxint;
    epoch:=0;
    pararead; {reading of network internal parameter}
    case inyc of {initialization of the weight values}
      1 : ran_initw; {random initialization}
      2 : read_weight_from_file;
    end;
  repeat

```

```

reset(inpf); {resetting of file to start read}
epoch_err_write; {display}
errt:=0;
repeat
    dataread; {data read from file}
    forwd; {forward propagation}
    control; {calculation of delta and error values}
    case method of {back propagation methods}
        1 : backprop;
        2 : back_moment;
    end;
until eof(inpf); {check of end of file}
if errt < 1000000
    then begin wmin:=w;
        errmin:=errt;
    end;
until (errt<deserr) or (ans2='S'); {check whether we reach the desired
accuracy}
write('The minimum error is = ',errmin);
w:=wmin; {minimum error causes weights}
wwrite; {writing of weight values to an output file}
end;
procedure test;
var testf,testoutf : string;
    tf,tof : text;
begin
testf:='xort.txt';
testoutf:='xortout.txt';
assign(tf,testf);
assign(tof,testoutf);
reset(tf);
rewrite(tof);
reset(outf);
for k:=1 to hlnum+1 do
    begin
        for j:=1 to hunum[k] do
            begin
                read(outf,wbias[k,j]);
                for i:=1 to hunum[k-1] do
                    read(outf,w[k,j,i]);
                readln(outf);
            end;
        readln(outf);
    end;
close(outf);
repeat

```

```

for i:=1 to inum do
begin
  read(tf,inpvec[i]);
end;
for k:=1 to hlum+1 do
for j:=1 to hunum[k] do
begin
  outvec[k,j]:=bias[k]*wbias[k,j];
  for i:=1 to hunum[k-1] do
    if k=1
      then outvec[k,j]:=outvec[k,j]+w[k,j,i]*f(funcnum1,inpvec[i])
      else outvec[k,j]:=outvec[k,j]+w[k,j,i]*f(funcnum2,outvec[k-1,i]);
    end;
  end;
for i:=1 to onum do
begin
  write(i:2,f(funcnum2,outvec[hlum+1,i]):15:9);
  write(tof,f(funcnum2,outvec[hlum+1,i]):15:9);
end;
writeln;
writeln(tof);
readln(tf);
until eof(tf);
close(tf);
close(tof);
end;
begin
write('Enter the input file name --> ');
readln(inpn);
write('Enter the output file name --> ');
readln(outn);}
inpn:='xor.txt';
outn:='wxor.txt';
assign(outf,outn);
write('Enter the hidden layer number--> ');
readln(hlum);
for i:=1 to hlum+2 do
begin
  if i=1
    then begin
      write('Enter the input unit number--> ');
      readln(inum);
      hunum[i-1]:=inum;
    end
    else if i<>hlum+2
      then begin

```



```

        write('Enter the',i-1,'.hidden unit number--> ');

        readln(hunum[i-1]);
    end
else begin
    write('Enter the output unit number--> ');
    readln(onum);
    hunum[i-1]:=onum;
end;

end;
write('Do you want to use bias values (y/n) ->');
readln(ans);
case ans of
    'y','Y': for i:=1 to hlnum+1 do bias[i]:=1;
    else for i:=1 to hlnum+1 do bias[i]:=0;
end;
writeln(' :20,1 - Identity');
writeln(' :20,2 - Sigmoid');
writeln(' :20,3 - Hyperbolic Tangent');
writeln(' :20,4 - Square');
writeln(' :20,5 - Sinus');
writeln(' :20,6 - Arc Tangent');
writeln(' :20,7 - Logaritmik');
writeln(' :20,8 - Gaussian');
write('Enter the Function number to use in input units-->');
readln(funcnum1);
write('Enter the Function number to use in hidden & output units-->');
readln(funcnum2);
repeat
    write(' What do you want to do ( {T}rain or t{E}st ) ->');
    readln(ans);
    case ans of
        'T','t' : train;
        'E','e' : test;
        else writeln('Wrong Entrance !!');
    end;
until not (ans in ['t','T','E','e']);
end.

```

APPENDIX C

NEURAL NETWORK CONSTRUCTION PHILOSOPHY

In this part of the study, the formulae used to construct the artificial neural network were given. The backpropagation algorithm, addition of momentum term, XOR problem results obtained from the developed neural network model could be found here.

C.1. Backpropagation Algorithm

Backpropagation (or the generalized delta rule) is a popularized supervised learning algorithm. It finally provided the multi-layer perceptron with an efficient learning rule. The step-by-step implementation of the backpropagation algorithm is as follows:

1. Weight initialization: Set the weights of all connections in the network to some real number in the range of $[-1,1]$,
2. Feed-forward stage: Present an normalized input pattern and corresponding desired outputs to the network,

3. Propagate the input vector forward through the network and then compute the output in response to the current input presented to the network,
4. Compute the difference between the network output and the desired ones,
5. Backpropagation stage: Starting from the output nodes, trace back to the hidden layers recursively. Update the weights between the j^{th} neuron in the layer of l and the i^{th} neuron in the previous layer by using the following formula:

$$\Delta w_{ji}^{n+1} = \eta y_i \delta_j^l + \alpha \Delta w_{ji}^n \quad \text{for } l=1, \dots, L \quad (1)$$

η : learning rate

α : momentum term

n : current iteration

y_i : output of the i^{th} neuron in the previous layer or input for the current j^{th} neuron in the layer of l ,

$$\delta_j^l = f'(\sum_k w_{kj}) - y_j \quad \text{for output layer neurons} \quad (2)$$

$$\delta_j^l = f'(\sum_k \delta_k w_{kj}) \quad \text{for hidden layer units} \quad (3)$$

6. the i, j, k represent previous, current and next layer neuron numbers respectively $f'(I_j)$ is the derivative of transfer function at n^{th} input value of j^{th} neuron, d_j is the desired output for j^{th} neuron, and y_j is the computed output of j^{th} neuron.

7. Weight update: Update the Weights in the network by using:

$$w_{ji}^{n+1} = w_{ji}^n - \Delta w_{ji}^{n+1} \quad (4)$$

8. Go to step2 and repeat the same steps until the global error is minimized.

C.2. Neuron Activation Functions

In this study 8 different functions to be taken into consideration (i.e. identity, sigmoid, hyperbolic tangent, square, sinus, arc tangent, logarithmic and gaussian function). Except for the gaussian function all of the function are continuous and differentiable everywhere.

C.3. Learning Rate and Momentum Term

The learning rate, η , determines how fast a network will learn the relationship between the input and the corresponding output patterns by controlling how much each input-output pattern influence the connections between the neurons. It is usually less than 1. If it is larger than 1 the learning is not stable but if it is too small, close to 0, stable learning can be achieved but very slow.

Sometimes a backpropagation network does not train in a reasonable period and the total error for the training set stop decreasing then the network is stuck in a local minimum. Adding the momentum term, α , into the network gives the ability to avoid such failures to the network and it works efficiently. It has a constant value between 0 and 1.

C.4. Exclusive OR (XOR) Function

The constructed ANN can be used to solve the famous XOR function given below:

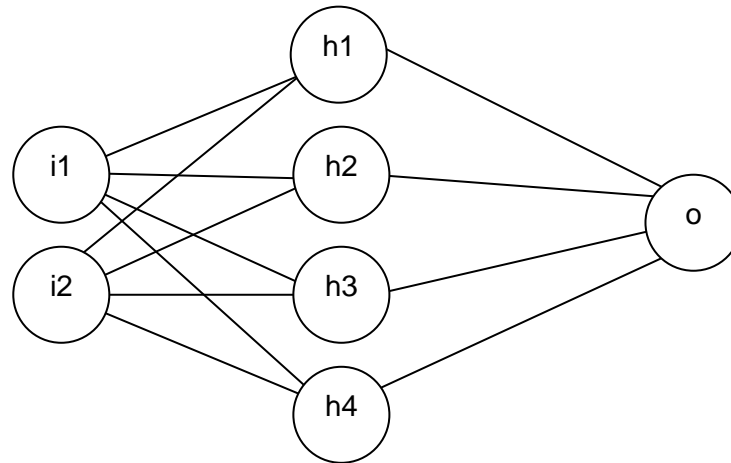


Figure C-1: A 2:4:1 network

Table C-1: The Famous XOR Function

i1	i2	output
1	1	0
1	0	1
0	1	1
0	0	0

The training of the sample network given in the Figure C-1 for the famous XOR problem given in the Table C-1 was completed by using the activation energy function of sigmoid. The testing data and the corresponding output data with epoches for the error values of 0.001 and 0.0001 for the simple backpropagation and the backpropagation with

momentum (the learning rate of 0.3 and the momentum of 0.7 values were chosen) were given in the Table C-2 and the Table C-3 respectively.

Table C-2: Testing Data and the Corresponding Output for 0.001 error

i1	i2	Backpropagation (epoch = 125332)	BPN with momentum (epoch = 39642)
1	1	0.014361832	0.014559412
1	0	0.974198879	0.974045794
0	1	0.975611253	0.975681649
0	0	0.022991430	0.022756205

Table C-3: Testing Data and the Corresponding Output for 0.0001 error

i1	i2	Backpropagation (epoch = 573839)	BPN with momentum (epoch = 174491)
1	1	0.003924480	0.003957929
1	0	0.992206968	0.992200086
0	1	0.992238511	0.992241421
0	0	0.007971149	0.007950123

Referring to above given data it can be easily seen that the momentum term makes the system speedier.

As a final notice, for the backpropagation with momentum, after 39642 epochs the network reached the training set error of 0.001 and the test set error of 0.01296 and also after 174491 epochs the training set error of 0.0001 and the testing set error of 0.00365 were reached. It can be said that from these comparisons after 174491 epochs, the developed network is able to respond to the XOR problem perfectly.

APPENDIX D

CT IMAGES

CT images of 6 different core plugs have been taken by using Philips Tomoscan 60/TX third generation scanner at CT Laboratory of Petroleum Research Laboratory at Middle East Technical University. Scan time of Tomoscan changes between 1.9 to 9 seconds with which 80 to 130 KeV energy can be utilized up to 700 mA tube current.

Length between adjacent slices has been set to 7 mm and in order to calculate porosity values of core plugs, serial scans have been carried out with two different energy levels (i.e. 100 KeV and 130 KeV) at 250 mA tube current.

CT images and calculated slice-wise porosity data plotted with the results obtained by helium porosimetry are as follows:

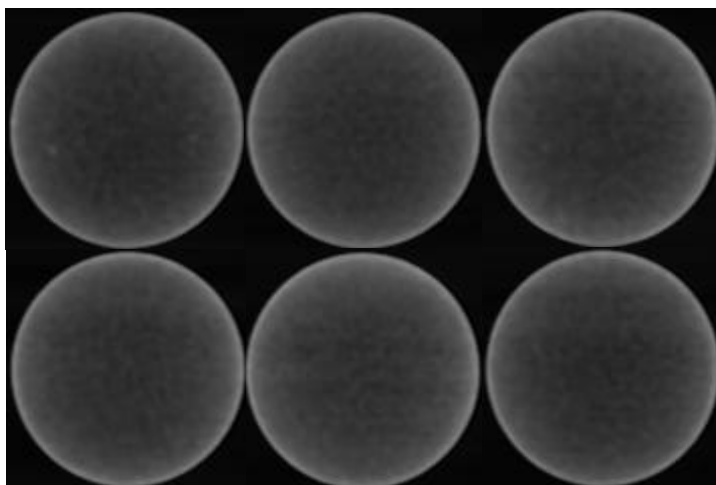


Figure D-1: CT images of Plug 4-1 (Sandstone)

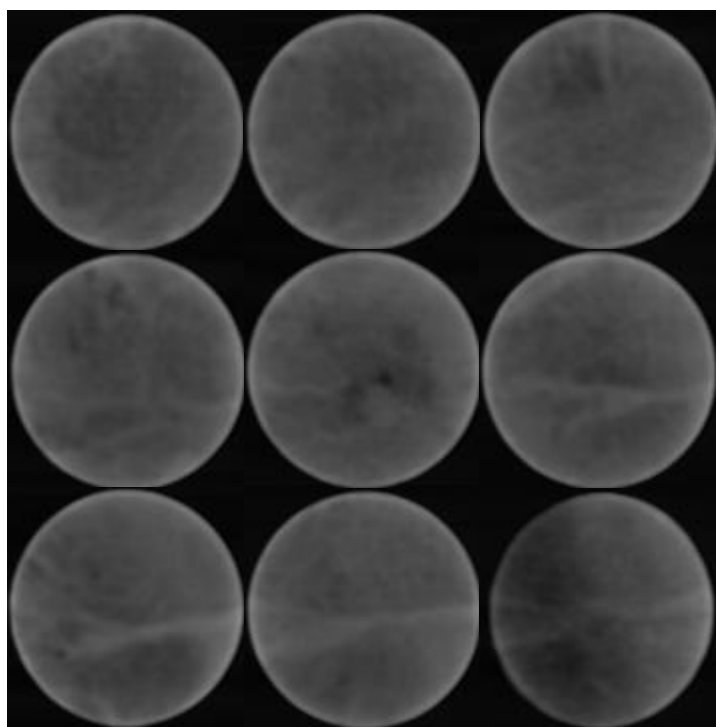


Figure D-2: CT images of Plug 92-2 (Dolostone)

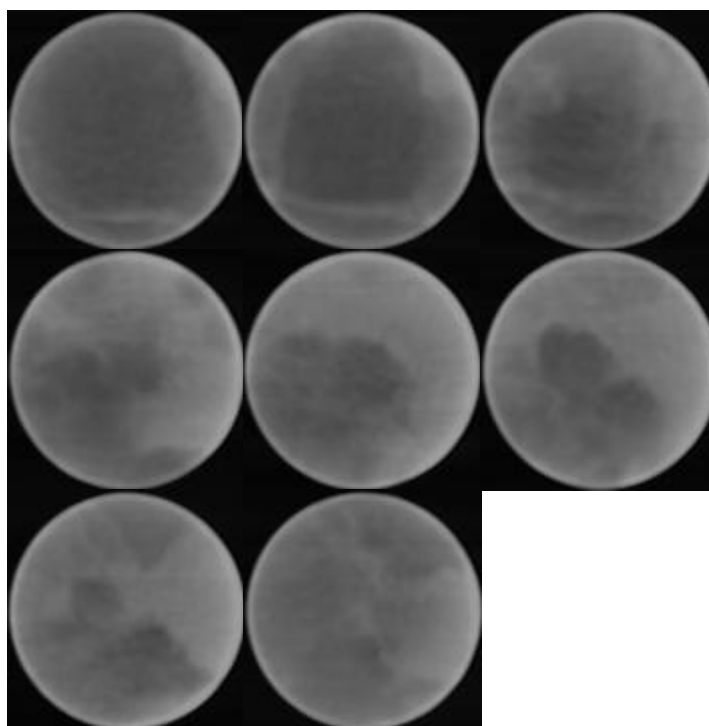


Figure D-3: CT images of Plug 100-4 (Dolomitic Limestone)

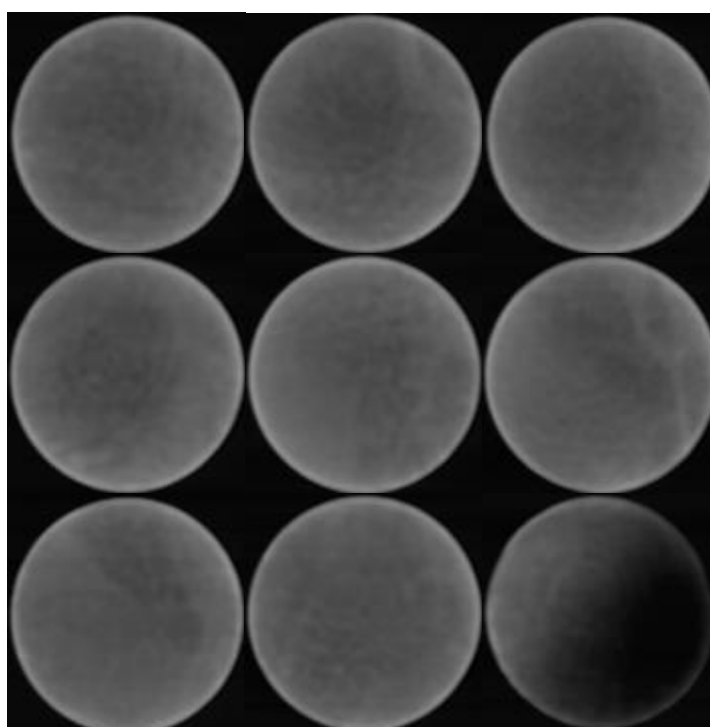


Figure D-4: CT images of Plug 91-5 (Dolostone)

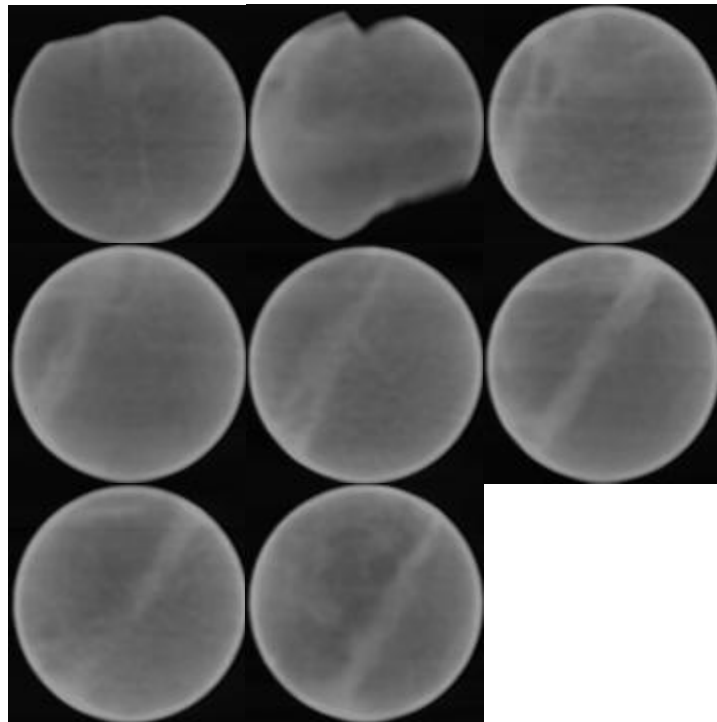


Figure D-5: CT images of Plug 83-6 (Dolomitic Limestone)

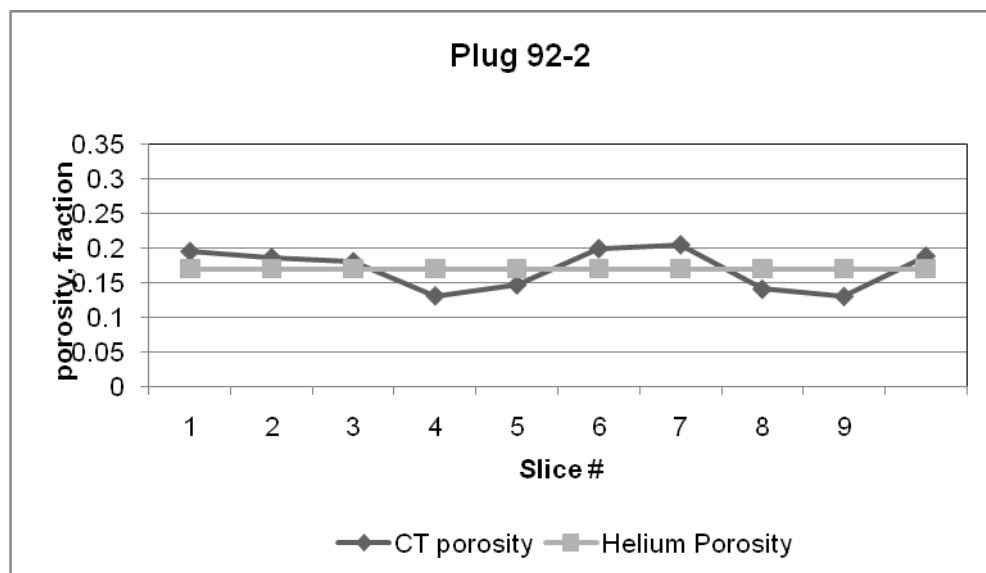


Figure D-6: Porosity Distribution in Plug 92-2 (Dolostone)

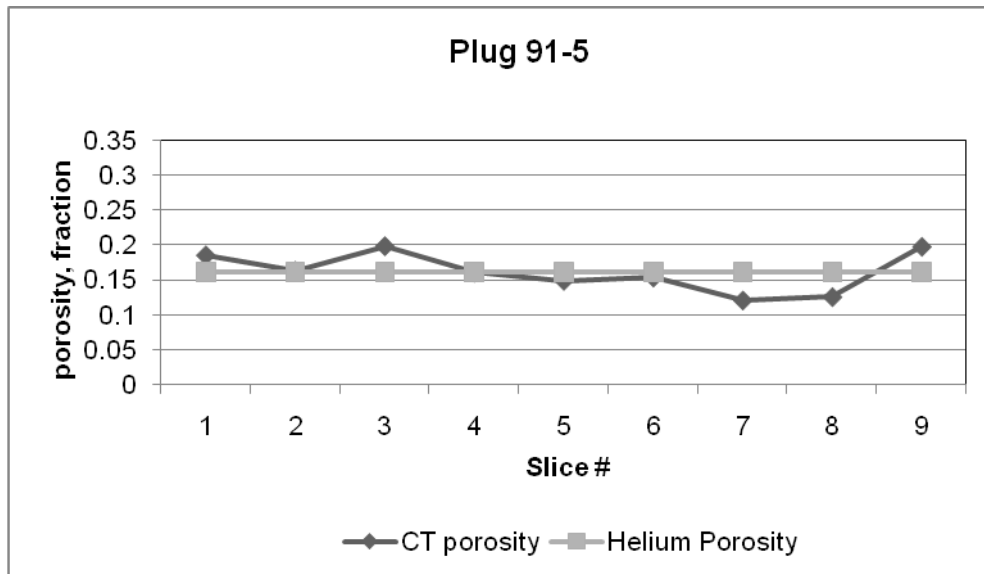


Figure D-7: Porosity Distribution in Plug 91-5 (Dolostone)

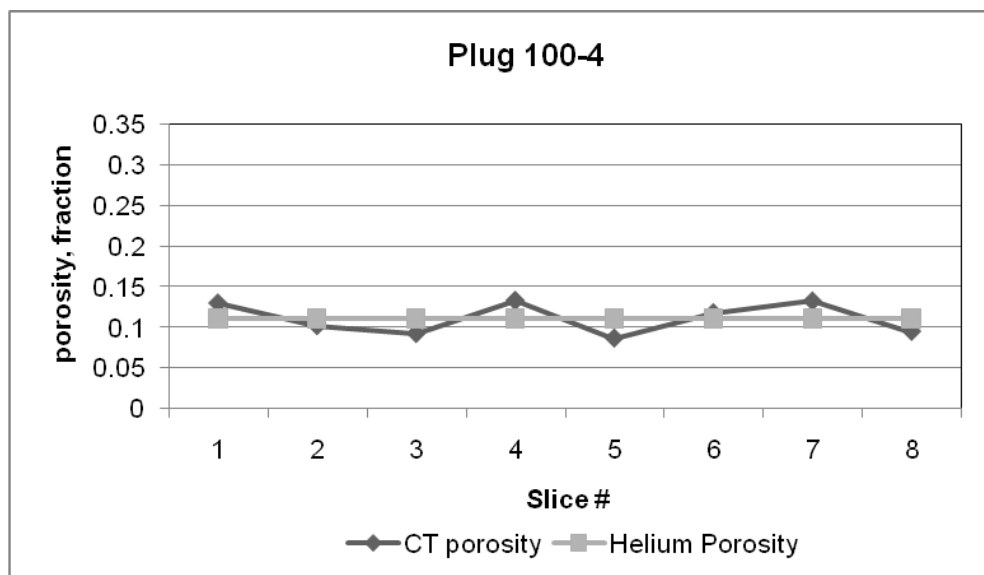


Figure D-8: Porosity Distribution in Plug 100-4 (Dolomitic Limestone)

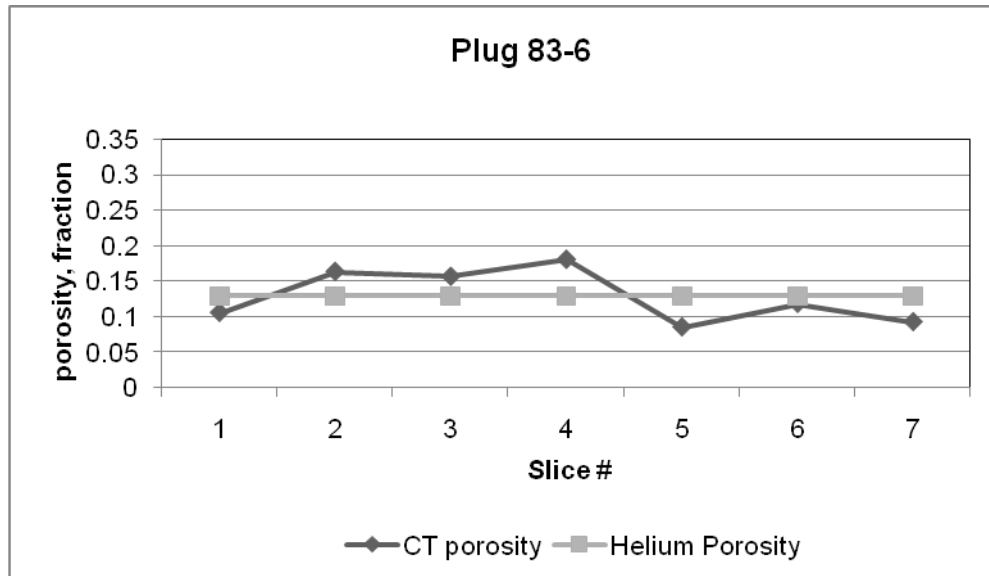


Figure D-9: Porosity Distribution in Plug 83-6 (Dolomitic Limestone)

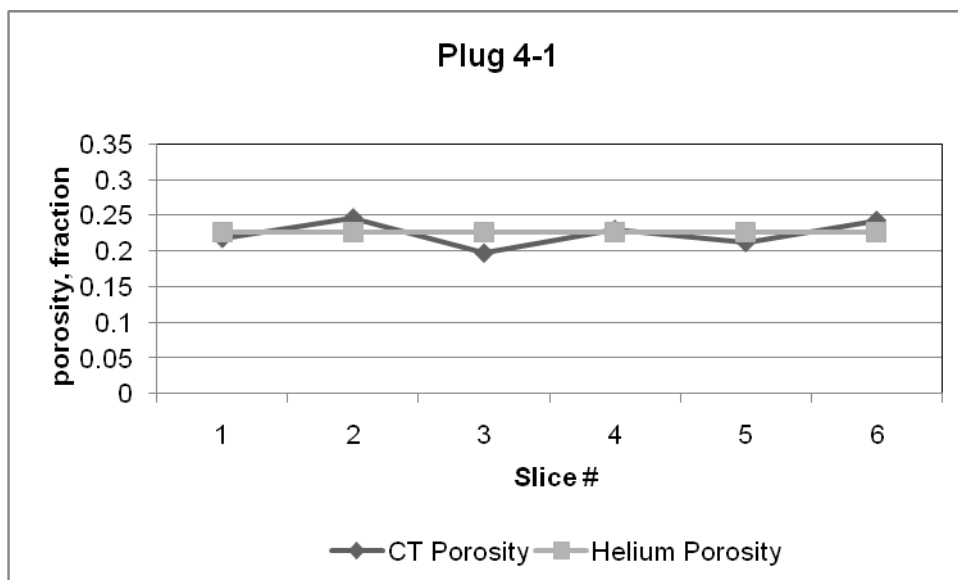


Figure D-10: Porosity Distribution in Plug 4-1 (Sandstone)

APPENDIX E

THIN SECTION IMAGES - POROSITY and PORE SIZE DISTRIBUTIONS

Seven serial thin sections had been made by Research Center at Turkish Petroleum Corporation (TPAO) on four different core plugs (plug 4-1, 92-2, 3-3 and 100-4) and nine (3x3) images had been taken on each thin section by using the Leica DFC 280 microscope available in the Image Analysis Laboratory of Petroleum Research Laboratory at Middle East Technical University. Pore body – throat size distributions and some of the thin section images have been given below.

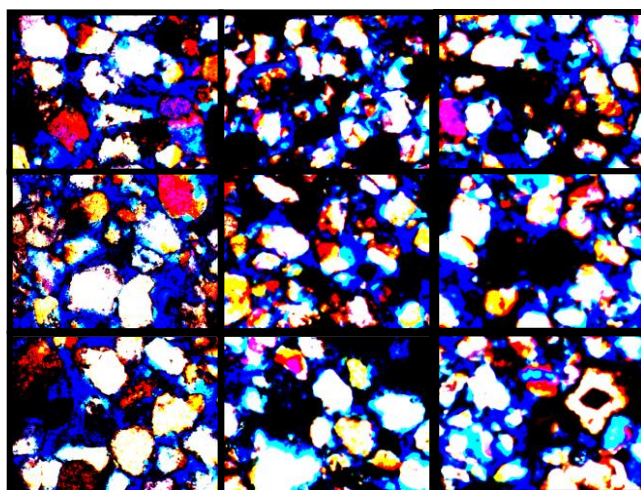


Figure E-1: 3x3 imaging of third thin section of Plug 4-1

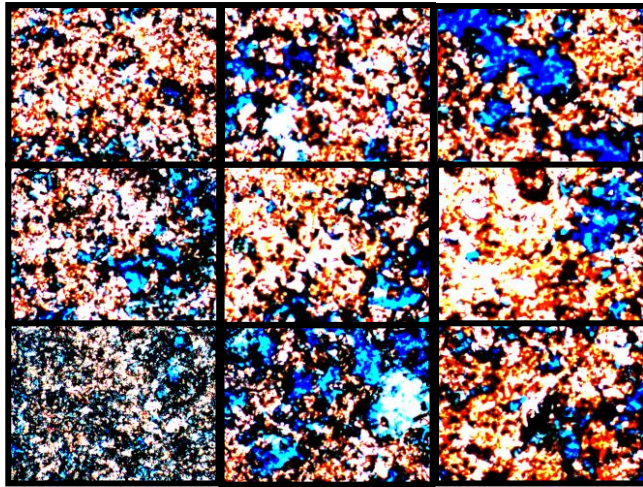


Figure E-2: 3x3 imaging of fifth thin section of Plug 92-2

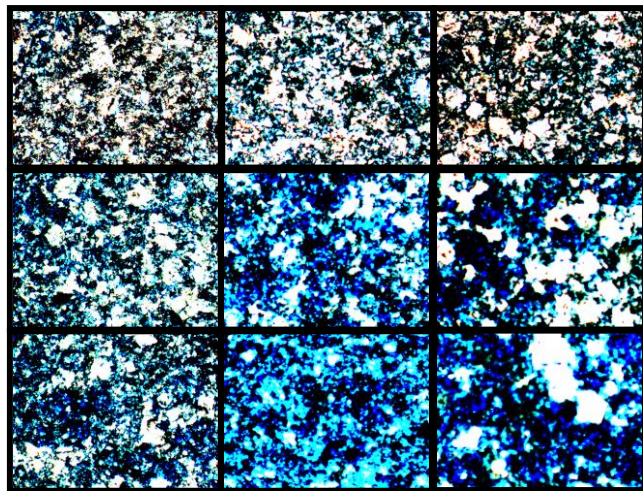


Figure E-3: 3x3 imaging of seventh thin section of Plug 100-4

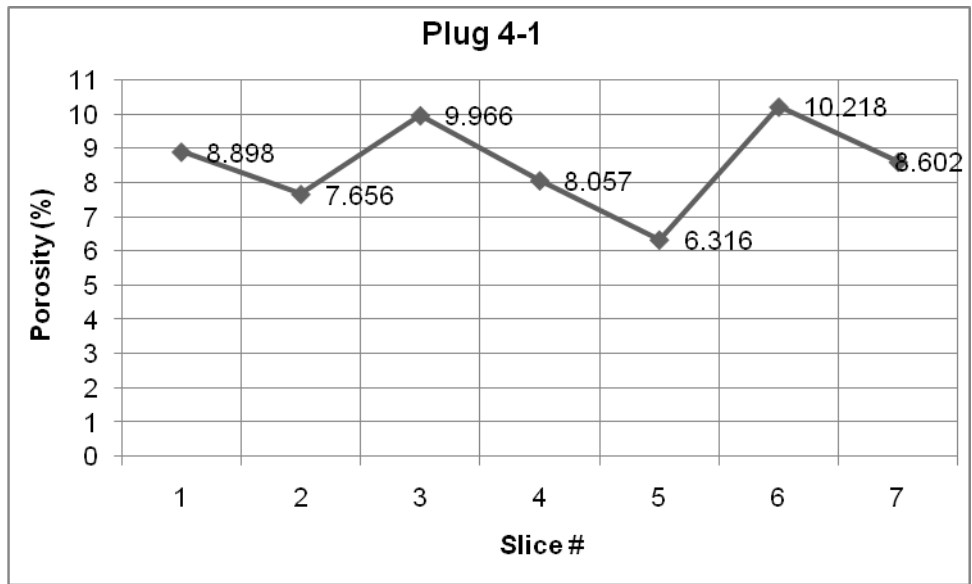


Figure E-4: Porosity Distribution for Plug 4-1 (sandstone) from Thin Section Imaging

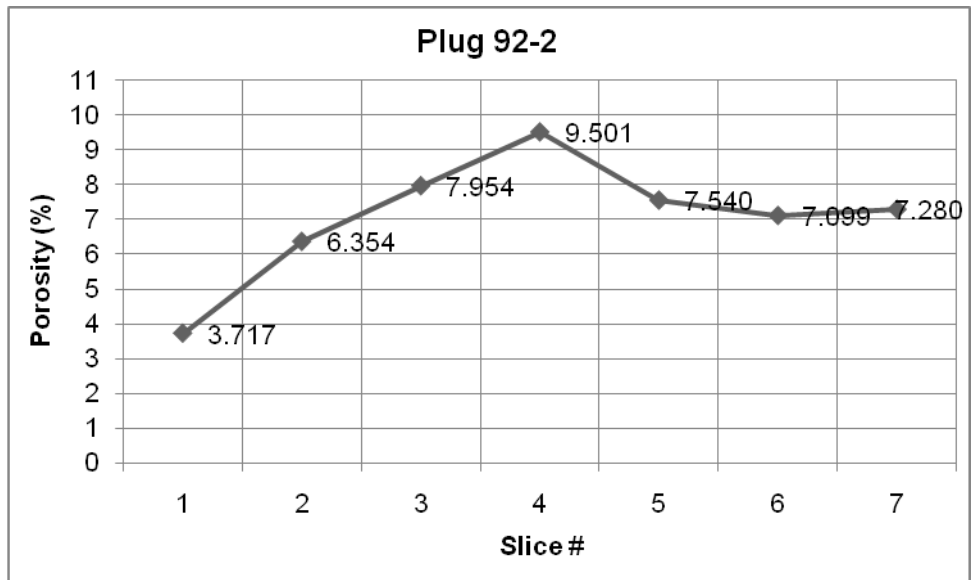


Figure E-5: Porosity Distribution for Plug 92-2 (Dolostone) from Thin Section Imaging

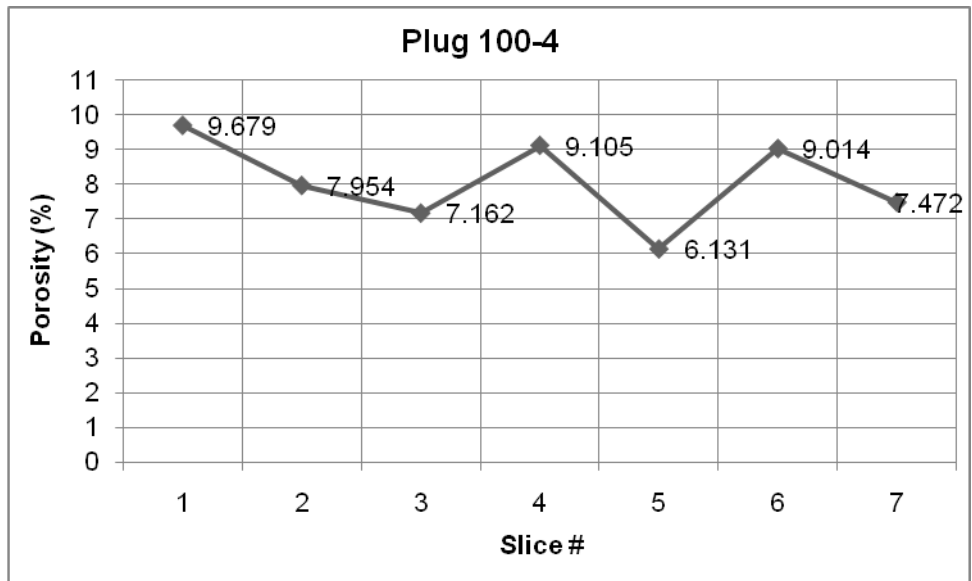


Figure E-6: Porosity Distribution for Plug 100-4 (Limestone) from Thin Section Imaging

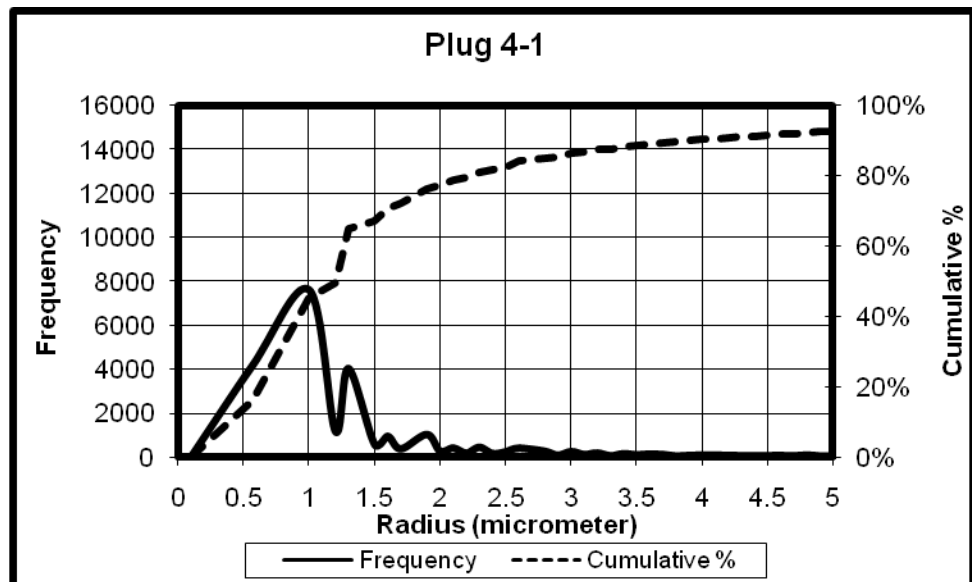


Figure E-7: Frequency & Cumulative % vs Radius (Plug 4-1)

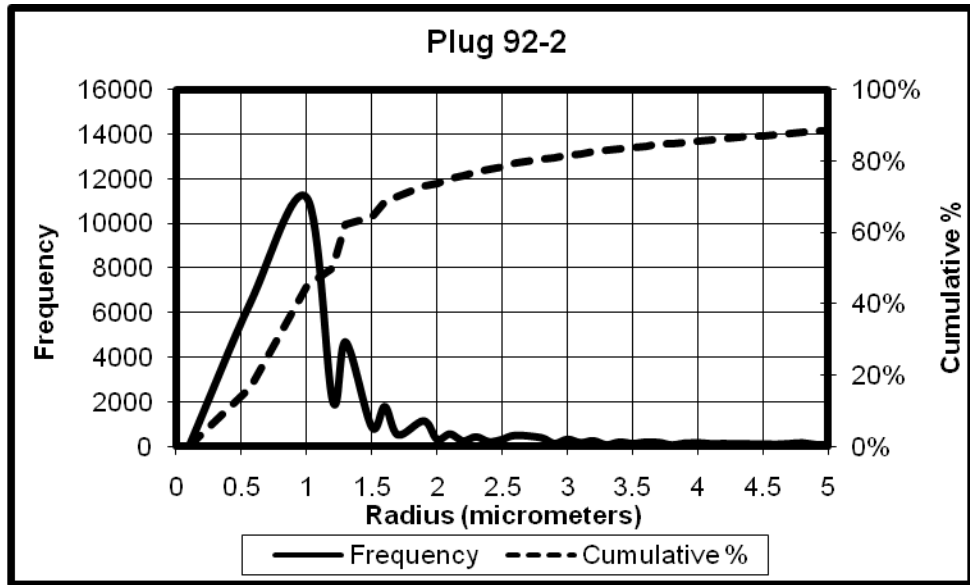


Figure E-8: Frequency & Cumulative % vs Radius (Plug 92-2)

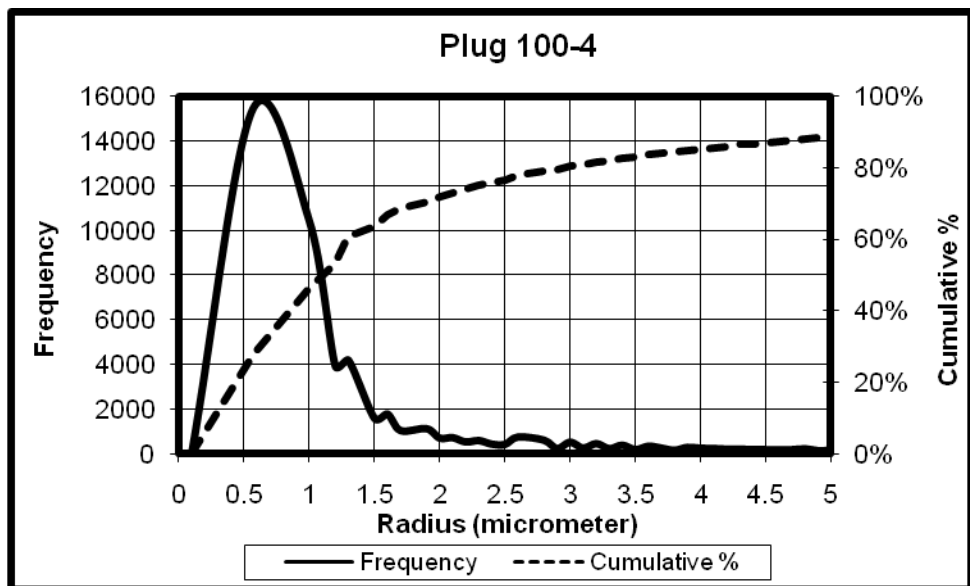


Figure E-9: Frequency & Cumulative % vs Radius (Plug 100-4)

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Karaman, Türker
Nationality: Republic of Turkey (T.C.)
Date and Place of Birth: 10 June 1975, Reşadiye - TOKAT
Marital Status: Married
Phone: +90 312 331 6228
Fax: +90 312 331 6198
GSM: +90 532 475 9382
Email: trkaraman@yahoo.com

EDUCATION

Degree	Institution	Year of Graduation
MS	METU, Petroleum and Natural Gas Eng.	2002
BS	METU, Petroleum and Natural Gas Eng.	1998
High School	Yüce Science High School, Ankara	1992

WORK EXPERIENCE

Year	Place	Enrollment
2006 – Present	EXPRO GROUP, Angola	Data Acquisition Eng.
2004 – 2006	TPAO, Ankara	Chief Reservoir Eng.
1999 – 2004	TPAO, Ankara	Reservoir Eng.

FOREIGN LANGUAGES

Fluent English

PUBLICATIONS

1. Karaman, T., Demiral, B. M. R., “Determination of Three Phase Relative Permeability Values by Using an Artificial Neural Network Model”, Energy Sources, Vol. 26, Num. 10, Aug 2004, pp. 903-914.
2. Karaman, T., Demiral, B. M. R., “Determination of Two and Three Phase Relative Permeability Values by Using a Pore Network Model”, Energy Sources, Vol. 26, Num. 7, June 2004, pp. 685-696.
3. Karaman, T., Demiral, B. “Two and Three Phase Relative Permeability Estimations by Using Pore Network Modeling”, paper presented at 14th International Petroleum and Natural Gas Congress and Exhibition of Türkiye, May 12-14, 2003, Ankara, Türkiye, pp. 322-332.

4. Karaman, T., Demiral, B. “Three Phase Relative Permeability Estimation by Using Artificial Neural Networks”, paper presented at 14th International Petroleum and Natural Gas Congress and Exhibition of Türkiye, May 12-14, 2003, Ankara, Türkiye, pp. 333-343.

5. Karaman, T., Erdogan, I., Demiral, B. “Relative Permeability Estimation by using Pore Network Modeling”, paper presented at 12th International Petroleum and Natural Gas Congress and Exhibition of Türkiye, October 12-15, 1998, Ankara, Türkiye, pp. 671-686.

HOBBIES

Playing Chess, Sightseeing, Computer Programming.