

## ABSTRACT

GHOSH, SUPRIYO. Simulation, Network Modeling, and Imaging of Porous Media Drying. (Under the direction of Yong Pan and Kevin M Keener.)

Drying is a unit operation where water is removed from a material by means of evaporation and mass transfer. It is widely used in industrial manufacturing processes. Many of these processes involve drying of porous materials. Porous media are, in general, heterogeneous systems. The microstructure of the pore space influences transport properties and hence, drying rates. Thus, a quantitative geometrical characterization of the pore space is crucial for accurate prediction of porous media drying rates. Simulation of porous media structure, given a specific size distribution of constituent particles, followed by pore-space characterization is a powerful as well as economical predictive tool for product design. Realistic unconsolidated porous media were reconstructed through Monte Carlo gravitational particle packing simulation. The porous media simulation was validated by comparison with extracted transport related micro-structural parameters from x-ray micro-CT (computed tomographic) images. A mathematical morphology based three-dimensional image processing algorithm was developed to characterize the pore space in the simulated porous media. This realistic pore-throat network information was utilized in an invasion percolation based simulation of porous media drying and showed satisfactory agreement with quantitative data extracted from in-situ NMR (nuclear magnetic resonance) imaging experiments.

**SIMULATION, NETWORK MODELING, AND IMAGING OF POROUS MEDIA  
DRYING**

by  
**SUPRIYO GHOSH**

A dissertation submitted to the Graduate Faculty of  
North Carolina State University  
in partial fulfillment of the  
requirements for the Degree of  
Doctor of Philosophy

**FOOD SCIENCE**

Raleigh, North Carolina

2007

**APPROVED BY:**

---

Brian E Farkas

---

John H van Zanten

---

Kevin M Keener  
Co-chair of Advisory Committee

---

Yong Pan  
Co-chair of Advisory Committee

## DEDICATION

I would like to dedicate this PhD research work and dissertation to my father, my mother and my sweet and loving wife, Payel.

## BIOGRAPHY

Supriyo Ghosh was born in a very small town, Purulia in West Bengal, India. He spent his childhood there and went to primary and high school. He did his Bachelor of Technology degree from Anand, Gujarat, India. Right after finishing his undergraduation he went to Utah State University, Logan, UT to start his Masters degree with a graduate scholarship in 2001. Upon finishing his MS from Utah State University, he joined North Carolina State University to start his PhD degree program in 2003. He is currently working as a scientist in an analytical instrument company, Bruker, and developing new applications and sensors of Nuclear Magnetic Resonance for process industries.

## ACKNOWLEDGEMENTS

I hereby acknowledge the great help I received from my advisors, Dr Yong Pan and Dr Kevin M Keener, and my other committee members, Dr Brian E Farkas and Dr John H van Zanten. I also acknowledge the help I have received from Dr Jonathan C Allen, and Ms. Susan Kall every time I had any administrative problem. I would also like acknowledge the following people in Procter & Gamble Co. who have helped me in many invaluable ways all along my PhD research: Mr. Michael J Bills, Ms. Paula Chmielewski, Mr. Tom Dufresne, Ms. Ying Biossy, Mr. Bill Hood, Dr John Hecht, Ms. Susan Forest, Ms. Elizabeth Picos.

## TABLE OF CONTENTS

	Page
LIST OF FIGURES .....	vii
INTRODUCTION .....	1
LITERATURE REVIEW .....	4
The Drying Process .....	4
Porous Media Drying Studied using Magnetic Resonance Imaging (MRI) .....	5
Modeling of Porous Media Drying .....	6
Invasion Percolation Modeling (IPM) .....	10
Three Dimensional Image Processing for Pore Network Extraction .....	13
The Physics of Magnetic Resonance Imaging .....	19
X-ray Microtomography .....	39
References .....	40
SIMULATION OF UNCONSOLIDATED POROUS MEDIA AND PORE-SPACE GEOMETRY CHARACTERIZATION .....	50
Abstract .....	50
Introduction .....	51
Methods .....	55
Results & Discussions .....	60
Conclusions .....	64
Acknowledgements .....	64
Literature Cited .....	65
A SIMULATION BASED METHOD TO ASSESS INVERSION ALGORITHMS FOR TRANSVERSE RELAXATION DATA .....	80
Abstract .....	80
Introduction .....	80
Methods .....	83
Results .....	85
Discussion .....	90
Acknowledgements .....	92
References .....	92
APPENDICES .....	94
STUDY OF POROUS MEDIA DRYING USING SIMULATED POROUS MEDIA, PORE NETWORK DRYING, AND MAGNETIC RESONANCE IMAGING .....	95
Abstract .....	95

Introduction.....	95
Methods.....	103
Results & Discussion .....	113
Conclusion .....	121
Acknowledgements.....	121
Literature Cited .....	121
DRYING SET-UP IMAGES .....	126
MATLAB CODES.....	131

## LIST OF FIGURES

	Page
CHAPTER II	
Figure 1: The three stages of drying .....	4
Figure 2: Nuclear excitation.....	25
Figure 3: Signal acquisition and fourier transformation .....	25
Figure 4: Spin-lattice or longitudinal relaxation process .....	29
Figure 5: Spin-spin or transverse relaxation process .....	29
Figure 6: Spin echo pulse sequence used to acquire a one-dimensional signal profile	33
Figure 7: Frequency encoding in one dimensional imaging .....	33
Figure 8: The slice selection process .....	35
Figure 9: Spin echo pulse sequence for two dimensional imaging.....	35
Figure 10: Spin echo two-dimensional imaging in k-space.....	36
Figure 11: Gradient echo pulse sequence for two-dimensional imaging.....	38
Figure 12: Gradient echo two-dimensional imaging in k-space .....	38
CHAPTER III	
Figure 1: Monte Carlo (MC) gravitational particle packing simulation algorithm .....	69
Figure 2: Example of movement of particles undergoing MC gravitational simulation	70
Figure 3: Normalized total potential energy of the particle system as the simulation progresses.....	70
Figure 4: An example of acquired microscopic images of glass beads .....	71
Figure 5: Size distribution of 500 $\mu$ diameter glass beads used in the experiment.....	71
Figure 6: Size distribution of 250 $\mu$ diameter glass beads used in the experiment.....	71
Figure 7: Flow chart of three-dimensional skeletonization algorithm to extract pore network from simulated unconsolidated porous media .....	72

Figure 8: Explanation of the definition of tortuosity used in the manuscript .....	73
Figure 9: (a) Cubic lattice arrangement of simulated bead pack(on the left); (b) three-dimensional skeleton extracted using current algorithm(on the right) .....	73
Figure 10: (a) Volume rendering of a Monte Carlo simulated reconstructed porous medium (on the left); (b) Volume rendering of the pore-space (on the right).....	74
Figure 11: Influence of particle size distribution width on bulk porosity obtained through simulation. The filled circle corresponds to the real pack of 250 $\mu$ diameter beads, y-error bar indicates standard deviations from triplicate simulations.....	74
Figure 12: Influence of particle size distribution width on coordination number obtained through simulation. The filled circle corresponds to the real pack of 250 $\mu$ diameter beads, y-error bar indicates standard deviations from triplicate simulations.....	75
Figure 13: Influence of mean particle size (with constant variance of 1.0) on bulk porosity obtained through simulation .....	75
Figure 14: Influence of mean particle size (with constant variance of 1.0) on coordination number obtained through simulation .....	76
Figure 15: Volume rendering of x-ray microtomographic image of random packing of glass beads .....	76
Figure 16: Three-dimensional rendering of a sample extracted pore-network.....	77
Figure 17: Pore size distributions obtained from simulated packed beds and $\mu$ -CT image of 250 $\mu$ beads. The y-error bars indicate the standard deviations in triplicate simulations .....	77
Figure 18: Cumulative distribution functions of pore size distribution of 250 $\mu$ bead pack; from packing simulations and $\mu$ -CT image. This is used in the Kolmogorov-Smirnov statistics calculation .....	78
Figure 19: Pore size distributions obtained from simulated packed beds and $\mu$ -CT image of 500 $\mu$ beads. The y-error bars indicate the standard deviations in triplicate simulations .....	78
Figure 20: Cumulative distribution functions of pore size distribution of 500 $\mu$ bead pack; from packing simulation and $\mu$ -CT image. This is used in the Kolmogorov-Smirnov statistics calculation .....	79

## CHAPTER IV

Figure 1: Comparison of true T2 distribution and UPEN computed T2 distributions from data with varying noise levels.....	86
Figure 2: Effect of signal-to-noise ratio in raw data on deviation of computed mean T2 value from the true mean T2 value .....	87
Figure 3: Deviation in computed T2 distribution width due to signal-to-noise ratio ..	87
Figure 4: Resolving efficiency of UPEN algorithm (while signal-to-noise is kept constant).....	88
Figure 5: Effect of signal-to-noise ratio on the peak resolving efficiency .....	89
Figure 6: Evolution of Kolmogorov-Smirnov statistics between true distribution and computed T2 distribution as starting distribution width is gradually changed keeping the signal-to-noise ratio constant .....	90

## APPENDICES

Figure 1: Microscopic images of contact angles before and after surface treatment.	103
Figure 2: Drying cell for in-situ drying inside MRI instrument .....	105
Figure 3: Controlled air drying set-up for in-situ drying of glass bead pack.....	105
Figure 4: Invasion percolation algorithm for drying simulation.....	109
Figure 5: Flow chart of algorithm developed for determination fractal dimension using box counting technique .....	111
Figure 6: Determination of fractal dimension using box counting method .....	112
Figure 7: Proton density map of glass bead pack obtained through NMR spin-echo micro-imaging (all time points are shown in the appendix section).[sample dimension: 1cmx1cmx2cm; scales indicate water proton density].....	117
Figure 8: Evolution of wet patches during 500 $\mu$ mean diameter glass bead drying (dotted line experimental data, solid line simulation data) .....	118
Figure 9: Evolution of wet patches during 250 $\mu$ mean diameter glass bead drying (dotted line experimental data, solid line simulation data) .....	118
Figure 10: Change in fractal dimension during 500 $\mu$ mean diameter glass bead drying (dotted line experimental data, solid line simulation data) .....	119

Figure 11: Change in fractal dimension during 250 $\mu$ mean diameter glass bead drying (dotted line experimental data, solid line simulation data) .....	119
Figure 12: Drying curves from gravimetric measurement and simulation for 500 $\mu$ diameter beads .....	120
Figure 13: Drying curves from gravimetric measurement and simulation for 250 $\mu$ diameter beads .....	120
Figure 14: Set up for in-situ drying inside MRI instrument .....	126
Figure 15: Drying columns, bypass line through humidifier, flow meter/controller.	127
Figure 16: PID controller, heat exchanger, online temperature sensor, Bruker Ultrashield 500MHz microimaging system .....	128
Figure 17: Drying Cell (a) top view (top image); (b) side view (bottom image) .....	129
Figure 18: Proton density map of glass bead pack obtained through NMR spin-echo micro-imaging .....	130
Figure 19: Calibration of microscopic images .....	131

## CHAPTER I

### **Introduction:**

Drying is a process in which water in a substance is vaporized and subsequently removed. It is frequently encountered as one of the unit operations incorporated into the processing sequence of various manufacturing industries e.g. food processing, chemical technology, textile, etc. The drying behavior of a material dictates its processing as well as functional attributes. It also dictates what the optimum parameters for a drying operation should be set at in order to make the process most economical. The drying pattern also determines what the texture and other mechanical properties of the dried material might be. Most of the solid materials that undergo drying in these manufacturing sectors are porous in structure. Hence, the ability to accurately describe the process of drying is necessary in having more control over the process and achieving the desired product characteristics in an efficient manner. Modeling porous media drying has been an active field of research for decades. However, even the simplest situation, such as the drying of a bed of glass beads at room temperature, is still a challenging mathematical problem to develop and solve. Traditionally, the problem is treated within the framework of a continuum approach to porous media.<sup>1</sup> This approach relies on the generalized Darcy's law and the relative permeability concept. In the process, the heterogeneous multiphase system of porous media is considered as a fictitious continuum media. As discussed by Masmoudi et al.<sup>2</sup>, the relevance of this traditional approach has not yet been fully demonstrated. The shortcomings of the continuum approach such as its inability to lead to a general method of predicting the interfacial mass transfer coefficients for drying, inspired the investigation of drying with a microscopic perspective: percolation modeling.<sup>3</sup> By modification of the invasion percolation

algorithm for drying, pore-scale drying in two-dimensions was simulated, and it was supported satisfactorily with simplistic experimental data.<sup>4,5</sup> But the percolation process in two-dimensional structure differs dramatically from that in three-dimensions; and the real-life porous media are all made of three-dimensional pore structure. An attempt has been made to extend this pore-network based percolation simulation to three-dimension. But these simulations are based on artificially fixed parameter distributions to characterize the pore structure.<sup>6</sup> Also, in these simulations, pores have been assumed to be at the corners of a perfect cubic lattice, and they are connected by straight throats.<sup>6</sup> These assumptions are far from the real porous media encountered in the industry. Hence these simulation results cannot satisfactorily describe or predict porous media drying. Building a realistic porous bed using simulations and then extracting the pore network information for use in the invasion percolation algorithm renders the method more close to actual porous media drying. A non-invasive, high resolution method can be used to probe the evolution of the gas-liquid interface during drying of porous media and validate the simulation results. Powerful image processing pipelines utilizing mathematical morphology based three-dimensional skeletonization and template based dilation can be used to extract the pore network parameters from the three-dimensional data. Hence, the relevant approach in understanding the drying of porous media in greater detail is to:

- reconstruct porous media with control over the grain size distribution using a simulation close to real-life situation;
- extract parameters of the simulated pore-space using image processing sequence;
- simulate drying using the extracted parameters with the help of invasion percolation modeling.

The particle packing simulation results generated in this research were validated by high resolution x-ray microtomographic imaging. The invasion percolation based drying simulation was supported by the quantitative analysis of the phase evolution images acquired non-invasively by magnetic resonance imaging.

## CHAPTER II

### Literature Review

#### *The Drying Process*

The process of evaporation of moisture in a material, followed by its removal at the surface is generally referred as drying. The surface area of the substance undergoing drying and the boundary conditions of the process e.g. air flow rate, temperature, relative humidity, pressure, etc. dicatates the mass transfer rate at the surface. The transport of moisture from inner portions of the drying material to its surface is influenced by the physicochemical properties (e.g. microstructure, hydrophilicity, etc.) of matrix, extent of saturation, temperature, etc. Wannanen et al.<sup>7</sup> provided a comprehensive review of various theories proposed to explain the mechanism of drying. During a drying process, initially the surface moisture is removed by the air flowing over the material. As surface moisture is removed moisture migrates from inside to the surface to replenish the saturation at the surface. During this first period the drying rate remains constant. As the drying continues, the flux of water from inside to the surface of the drying material, driven by the capillary force, becomes less than the surface evaporation rate, and the surface starts to become partially dry reducing the evaporation area. This leads to first falling rate period of drying.

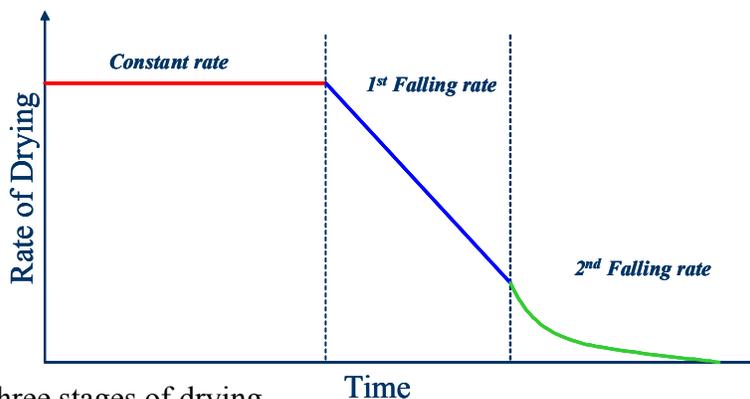


Figure 1: The three stages of drying

As the whole surface of the material becomes dry, the drying front starts to move inside the material. This movement is greatly influenced by the matrix property and leads to the second falling rate of drying. This process continues until the saturation of the whole matrix come to equilibrium with the air flowing over it, and then the drying process stops. In order to quantitatively study the evolution of the drying front some researchers have tried invasive methods. They obtained one-dimensional profiles of drying by slicing the material at different time points during the drying process.<sup>8</sup> This method cannot obtain high resolution, and also inaccuracy in quantification is introduced due to drying of the sample during the process of slicing and gravimetric measurements.

#### ***Porous Media Drying Studied using Magnetic Resonance Imaging(MRI)***

MRI is a noninvasive technique used in the field of drying. Waananen et al.<sup>7</sup> have suggested that MRI may be one of the best new tools for validating mathematical models of the drying process as it provides an accurate measurement of the internal moisture profiles in samples during drying. MRI techniques have been employed to study the drying of a variety of organic and inorganic materials. Perez et al.<sup>9</sup> measured the moisture concentration profiles of an apple during drying. Ruan et al.<sup>10</sup> and Song and Litchfield<sup>11</sup> measured the moisture profiles in a potato cylinder and a model food gel, respectively. The authors also calculated the moisture diffusion coefficients at different moisture contents by fitting the experimental data to a Fickian model of diffusion. Hills et al.<sup>12</sup> measured the radial moisture profiles of rehydrated pasta during drying. Besides food materials, MRI has also been used on inorganic porous materials. Hollewand and Gladden<sup>13</sup> studied the drying of catalyst pellets using spatially un-resolved NMR and MRI techniques. From the measured  $T_1$ ,  $T_2$  and diffusivity of the catalyst pellet at different saturation levels, they concluded that the transport process

occurring during drying is very sensitive to the pore structure of material. Beyea et al.<sup>14</sup> acquired one-dimensional moisture concentration profiles of concrete during drying. Pel et al.<sup>15</sup> measured one-dimensional moisture concentration profiles of fired-clay brick, sand-lime brick and gypsum during drying and obtained an estimate of moisture diffusivity as a function of moisture content. Koptuyug et al.<sup>16</sup> acquired one-dimensional moisture concentration profiles of alumina and titania catalyst pellets and from these profiles they postulated the distribution of liquid in pore space at different stages of drying. In these previous studies, only spatially unresolved or one-dimensional information have been obtained using MRI techniques. De Barquin and Dereppe<sup>17</sup> obtained 2-D images of a white porous limestone at different stages of drying and heterogeneous distribution of water was observed from those images. Schrader and Litchfield<sup>18</sup> acquired 3-D images of an ear of corn during drying. However, the 3-D images were reconstructed from a series of 2-D images acquired using a multi-slice imaging method and the thickness of each slice was 1.3 mm. Recently Ruiz-Cabrera et al.<sup>19</sup> used 2D imaging of drying gelatine gel slabs to derive the 1D moisture profile in order to account for the shrinkage effect. They then used this data to investigate the dependence of diffusivity on saturation.

### ***Modeling of Porous Media Drying***

Models describing the process of drying aid in predicting the drying behavior of a porous media based on some knowledge of its physico-chemical characteristics, and the drying conditions. Thus, drying simulation can be used for process design, control and optimization of the profit. However, the development of any drying model is complicated due to the interaction between heat and mass transfer both on the surface of the medium and through its internal structure. Hence, mathematical modeling of drying of porous media has

been a major topic of research for many decades and a vast number of models are available in the literature. Waananen et al.<sup>7</sup> have presented a review on drying models that have been developed in the past. There are basically two different approaches in the modeling of drying: the traditional continuum approach, and the more recently considered discrete approach.

### *Continuum Approach*

The continuum approach considers the drying medium as a structureless continuum. Macroscopic partial differential equations are postulated by volume averaging the microscopic conservation laws. Phenomenological and empirical parameters are then used to find correlations between gradients and fluxes. Most authors adopt this method based on Whitaker's theory<sup>1</sup>. There are two different types of continuum models in the literature: the diffusive model and the capillary model.

The *diffusive models* assume that the diffusion is the only transport mechanism. As early as in 1933, Gilliland and Sherwood<sup>20</sup> had already modeled drying as a one-dimensional diffusive problem. This diffusive model is used by many other researchers including Dietl et al.<sup>21</sup>, Rogers and Kaviany<sup>22</sup>, Chen and Schmidt<sup>23</sup>, and Boukadida and Ben Nasrallah<sup>24</sup>. Dietl et al.<sup>21</sup> and Chen and Schmidt<sup>23</sup> developed models that account for the receding drying front in the falling rate period. Different sets of mass and heat transfer equations are developed for the wet and dry regions, separated by the evaporation front. The model proposed by Rogers and Kaviany<sup>22</sup> used the concept of relative permeability and the continuity and momentum equations given by Whitaker.<sup>25</sup> While all the models referred to above described only one-dimensional drying problem, Boukadida and Ben Narsallah<sup>24</sup> developed a numerical model for two-dimensional heat and mass transfer during convective drying porous media.

The *capillary model* assumes that capillary flow is the only transport mechanism during porous media drying. There are relatively fewer researchers using capillarity as the sole transport mechanism in drying models, among them are Comings and Sherwood<sup>26</sup> and Kiskurek and Gebizlioglu<sup>27</sup>. Comings and Sherwood<sup>26</sup> quantitatively accounted for the observed drying characteristics of a clay mix using a capillary moisture transfer mechanism. Kiskurek and Gebizlioglu<sup>27</sup> developed a model that assumes that the liquid moisture moves in a composite network of capillaries in the solid, and is carried to the surface by capillary force only. The shortcoming of this model was that evaporation is assumed to take place only at the solid surface and thus does not account for the receding drying front.

#### *Discrete Approach*

In general, there are two types of pore-scale modeling: Lattice-Boltzmann (LB) and pore network models. LB models, utilizing concepts originally introduced with lattice-gas models, are capable of simulating flow and transport in the actual pore space (Gunstensen and Rothman<sup>28</sup>; Rothman and Zaleski<sup>29</sup> and Dawson et al.<sup>30</sup>). Increasing computational power has allowed for more realistic LB simulations. However, these models are very complex and there is limited data for quantitative comparisons.

Pore network models are effective tools used to investigate or predict macroscopic properties from fundamental pore-scale behavior of processes and phenomena based on geometric volume averaging. Pore network models are mechanistic models that utilize an idealization of the complex pore space geometry of the porous media. This is usually achieved by representing the pore space by pore elements having simple geometric shapes. For example, pore-bodies and pore-throats have been represented as spheres and cylinders or cubes and cylinders, respectively. Network models have been used in the fields of chemical

engineering, petroleum engineering and hydrology to study a wide range of single and multiphase flow processes. They have been used to study relative permeability (Blunt and King<sup>31</sup>; Rajaram et al.<sup>32</sup> and Fischer and Celia<sup>33</sup>); the effect of pore structure on relative permeability and capillary pressure hysteresis in two phase systems ( Jerauld and Salter<sup>34</sup>); prediction of permeability and hydraulic conductivity ( Bryant et al.<sup>35</sup>); investigation of the functional relationship between capillary pressure, saturation, and interfacial areas ( Reeves and Celia<sup>36</sup>); prediction of permeabilities and residence time distributions for mechanical dispersion in packed beds ( Thompson and Fogler<sup>37</sup>); drainage and imbibition ( Lowry and Miller<sup>38</sup>); phase distributions, interfacial areas, and mass transfer ( Dillard and Blunt<sup>39</sup>); and ganglion formation and mobilization ( Dias and Payatakes<sup>40</sup>; Li and Wardlaw<sup>41,42</sup>).

However, there are only very limited studies using network modeling to investigate phase change phenomena like drying in porous media. Satik and Yortos<sup>43</sup> and Li and Yortos<sup>44</sup> studied bubble growth and nucleation as a result of pressure reduction or heat transfer. McCall and Guyer<sup>45</sup> simulated an equilibrium evaporation phenomenon on a square bond lattice by lowering the vapor pressure. Prat and co-workers (Prat<sup>46,47</sup>; Laurindo and Prat<sup>4,5</sup>; Le Bray and Prat<sup>48</sup>) modeled drying as an invasion percolation process in which the viscous force is assumed to be negligible compared to the capillary force. They performed simulations on two- and three-dimensional set-up, and also performed experiments in ideal two-dimensional set-up. They tried to qualitatively as well as quantitatively show the phase distribution and drying curves through this process. Agreement between the experimental and simulation-based studies was very good in two-dimensions but not as satisfactory while trying to explain the experimental data in real three-dimensional porous media. Several reasons can be put forth explaining the discrepancies observed. Two-dimensional simulation

of network modeling can never be close to the experimental results involving real porous media. This is because, the connectivity in two-dimensions is very different from that in three-dimensions, dramatically influencing the critical percolation values or percolation threshold. Thus using an experimental method to non-invasively look-into the three-dimensional phase distribution in the porous media during the time course of drying, extracting the quantitative data from those three-dimensional data can really assist develop the invasion percolation model of drying of porous media as well as evaluate the same.

### ***Invasion Percolation Modeling(IPM)***

#### *Invasion Potential*

In invasion percolation the invasion of sites at each time step of the simulation is decided by the invasion potential of the sites located at the interface between the invading and the defending fluids. A physical basis can be given to the magnitude of this invasion potential by relating it to the capillary pressure. The capillary pressure,  $\Delta P_{\text{cap}}$ , across a liquid-liquid interface in a tube is:

$$\Delta P_{\text{cap}} = 2(\sigma \cos \theta) / R \quad (1)$$

where,  $\sigma$  is the interfacial tension between the two fluids,  $\theta$  is the contact angle, and  $R$  is the radius of the tube.

There are two different immiscible displacement processes: imbibition and drainage. During imbibition, a wetting fluid displaces a non-wetting fluid. In the absence of viscous and buoyancy forces, capillary forces will result in spontaneous imbibition of the wetting fluid into the pore space. According to the capillary pressure equation above, the capillary forces are strongest at the narrowest places in the porous medium. Therefore, if all the pore throats are smaller than the pore bodies, the invasion of the pore body will lead to spontaneous

invasion of the adjacent pore throats. Hence, the radii of the pore bodies are the controlling factor in imbibition. The capillary pressure is larger in smaller pore bodies, and hence those will be invaded first. Thus, for imbibition, the invasion potential is directly proportional to the capillary pressure established in the pore bodies.

During drainage, a non-wetting fluid displaces a wetting fluid. In this case the capillary pressure established in the pore bodies and pore throats will resist the invasion of the non-wetting fluid. The capillary forces are stronger at throats than in pore bodies. The non-wetting fluid invades the pore bodies more easily than the pore throats and thus drainage is controlled by the pore throats, with the larger pore throats being more easily invaded. Hence, in case of drainage, the invasion potential is inversely proportional to the capillary pressure in the pore throats.

#### *Previous Research Works in Percolation Modeling*

Broadbent and Hammersley<sup>49</sup> first proposed the theory of percolation in the year of 1957. Percolation essentially describes the accessibility of the network or lattice. The sites in a lattice or the bonds between these sites can be either open or closed with respect to transport through them. IPM was first proposed by Chandler et al.<sup>50</sup> and first implemented by Wilkinson and Willemsen<sup>51</sup> on various regular two-dimensional and three-dimensional lattices both with and without trapping of the defender fluid. Each site was invaded one at a time after each time increment in the simulation, the site invaded being the one with the highest invasion potential of all sites at the liquid-liquid interface. Therefore, invasion percolation is essentially a dynamic form of the percolation theory. As noted by these authors, although IPM is motivated by the liquid-liquid displacement process, it is applicable to any invasion process following the path of least resistance. Wilkinson and Willemsen<sup>51</sup>

demonstrated that when trapping of the defending fluid was included a significant difference was revealed between the two-dimensional and three-dimensional lattices, the clusters of trapped defender fluid being considerably larger for the two-dimensional lattice. This result shows a limitation of the two-dimensional lattice to model realistic porous systems that are usually inherently three-dimensional. Wilkinson<sup>52</sup> and Glass and Yarrington<sup>53</sup> later extended the IPM simulation of Wilkinson and Willemsen to include the effect of gravity as the stabilizing and destabilizing force.

#### *Drying as an Invasion Percolation Process*

Shaw<sup>54</sup> first suggested that drying can be considered as an invasion percolation process. As drying is generally a slow process, capillary forces dominate over viscous forces. The air invades the pore throat with the lowest capillary pressure, i.e. the pore throat with the widest width. The pore body, normally wider than the pore throat, is automatically invaded once the adjacent pore throat is invaded. This forms the basis of the invasion percolation model of drying. Prat and his co-workers<sup>4,5,46,47,48</sup> recognized the similarities between drainage and drying, and started to model drying as an invasion percolation process. In both drainage and drying, the liquid within the medium is the defender fluid while the air is the invader fluid. However, the air and the liquid exit the medium in the same direction in drainage and in opposite directions in drying. Another major difference between drying and conventional invasion percolation is that in drainage, isolated clusters of liquid can become trapped and cannot be invaded further due to fluid incompressibility; clusters of liquid become isolated from the main bulk but gradually disappear under the action of evaporation. Prat and his co-workers considered these dissimilarities. They also adopted some reasonable simplifying assumptions: the liquid is a single component liquid, the gas phase is a binary

mixture of vapor of the liquid and inert air, Kelvin effect can be neglected, diffusive transport is quasi-steady, vapor escapes through the top edge of the network, and zero flux conditions prevail across the other three boundaries of the porous medium. Based on these Prat developed a drying algorithm which is able to simulate porous media drying.

### ***Three Dimensional Image Processing for Pore-Network Extraction***

The most critical part in constructing a pore network model is defining its structure<sup>55</sup> which includes determination of pore-body locations, pore-body size distributions, pore-throat size distributions, connectivity, and the spatial correlation between pore-bodies. Several methods have been used to obtain pore network structure from porous media systems, including: estimation of pore sizes from measured capillary pressure–saturation curves<sup>56,57</sup>; fitting generated pressure–saturation curves to the measured curves<sup>33</sup>; and obtaining the statistics of the network from three-dimensional characterization of the pore space based on two-dimensional images<sup>58</sup>. The problem with using the first two approaches is the non-uniqueness of the solution obtained due to the dependency of the retention curves on both the topology of the pore space and pore size distribution<sup>59</sup>. Also, no information about the connectivity of the pore space is obtained using these methods. Therefore, it is desirable to obtain network parameters from a three-dimensional representation of the pore space.

There are several approaches for obtaining the three-dimensional representation of the pore space, including: simulation of random close packing of porous media<sup>35,37</sup>; three-dimensional reconstruction of the pore space based on the measured porosity and two-

correlation function of serial cross-sections<sup>58,60,61,62,63</sup>; and direct non-destructive three-dimensional imaging<sup>64,65,66,67,68,69,70</sup>.

Network models developed from computer-generated (i.e. ideal) packings mainly restrict their application to ideal systems consisting of spherical grains. While useful for studying important pore-scale processes and mechanisms, the need to extend the modeling to more realistic porous media systems (i.e. natural) is obvious. Destructive imaging of serial cross-sections is a widely used tool used to provide more realistic representation of the porous media and hence the pore space. In this technique, three-dimensional images are reconstructed based on properties of the imaged cross-sections (e.g. n-point correlation function). One of the problems associated with this method is the need for detailed thin cross-section to obtain accurate characterization of the system. This is often impractical or impossible to achieve because of the laborious method of slicing and digitizing, its destructive nature, and the resolution dependency of thickness of the slices.

Non-destructive direct imaging approaches (e.g. microtomography and nuclear magnetic resonance imaging) are attractive because they provide the actual morphology of the pore space. Thus, the connectivity and spatial variation of the pore-bodies and throats are retained. Among these high resolution non-invasive methods, for extraction of three-dimensional quantitative description of porous media, nuclear magnetic resonance imaging has an edge over the rest. The computed tomography method, can generate very high resolution three-dimensional images, but it can only sense the solid matrix of the porous medium and maps the pore structure in a negative sense. But, nuclear magnetic resonance imaging directly probes the water occupying the pore spaces, and thus, using this method, it is possible to observe the dynamics of the fluid interface or the phase distribution, along the

drying time-course. The resolution of the three dimensional images produced by the higher end  $\mu$ MRI instruments is good. The 500MHz  $\mu$ MRI system present in the Miami Valley Innovation Center (Procter & Gamble Co. corporate R&D Chemical Technology Division) can generate three-dimensional images down to a resolution of  $50\mu\text{m}\times 50\mu\text{m}\times 100\mu\text{m}$ .

Several approaches may be utilized to obtain the topology and the geometry of the pore space from three-dimensional images.

#### *Binary Representation of the Pore Space*

Hazlett<sup>71</sup> used three-dimensional images of Berea sandstone obtained using synchrotron X-ray microtomography. These images were used as input in a network model to simulate drainage, imbibition, and fluid spatial distributions. In his work, the pore space was not partitioned into network elements in the form of pore-bodies connected to pore-throats; pixellization of the pore space was used instead. The model consisted of a map of inscribed spheres where each pixel in the pore space is assigned a number corresponding to the radius of the largest sphere centered in that pixel without intersecting the solid phase. This inscribed sphere map provides the framework for equilibrium saturation distributions based on a cut-off radius corresponding to capillary pressure. A drawback of using this approach is overestimating the phase volumes at high capillary pressure during imbibition because the mean curvature of convex shapes is not extracted due to the use of a map of spheres with concave curvatures.

#### *Multiorientation Scanning Algorithm*

Zhao et al.<sup>72</sup> developed a multiorientation scanning algorithm to partition the pore space to pore-bodies and pore-throats. This algorithm was then adopted by Ioannidis and Chatzis<sup>62</sup>. In this algorithm, the throats are identified by finding local minima along the pore

space pathways by scanning the microstructure from nine different orientations, three orthogonal and six diagonal orientations. Scanning from different orientation produces overlapping of identification of the local minima and then the smallest is considered to be the throat. To identify pore-bodies, pore-throats and solid voxels are clustered in one phase and the remaining void space is clustered to represent the pore-bodies. A limitation with this approach is its inability to define inscribed pore-bodies.

#### *Morphological Thinning Algorithm*

Baldwin et al.<sup>68</sup> developed a morphological thinning algorithm to obtain a representation of the skeleton (medial axis) of the pore space and partition the void space into corresponding pore-bodies and pore-throats. No attempt was made to use the medial axis itself as a tool to characterize pore-bodies and pore-throats; the thinned images were used instead. In their work, pore-bodies are identified by summation of the voxels bounded by the solid matrix and locations corresponding to local minima are defined as pore-throats. A limitation of this approach is that the thinning algorithm is based on a simple segmentation algorithm that leads to misidentification or less accurate partitioning of the pore space.

#### *Medial-axis Skeletonization*

Lindquist and Venkatarangan<sup>69</sup> developed algorithms to calculate effective pore-body and pore-throat radii and other geometric properties such as interconnectivity based on the medial axis of the pore space. Their algorithms primarily have been applied to consolidated systems, and they themselves mentioned that it is not clear whether these algorithms are applicable to higher porosity materials (e.g. unconsolidated media).

#### *Quantitative Morphology*

Vogel and Roth<sup>61</sup> generated a pore network model based on pore-body size distribution and connectivity function obtained by calculating three-dimensional Euler number. Pore-body size distribution is obtained by a series of erosion dilation algorithms. In order to determine the proportions of the pores with a radius smaller than a given radius  $r$ , a sphere of a radius  $r$  is placed in every location in the pore space. Then erosion is performed to enclose all pore voxels where the sphere fits completely into the pore space. This step is followed by dilation to remove all pores smaller than  $r$ . The connectivity within and between different classes of pore sizes is determined by a specific Euler number which is a function of number of vertices, edges, faces, and volumes of the binary image which represent the pore space. A limitation of this algorithm is that the range of the pore sizes should be known a priori.

On the basis of these previous efforts and their outcomes it seems rational to extract the quantitative information from the three-dimensional nuclear magnetic resonance images using the following scheme: segmentation of the image into pore-space and solid matrix by converting the gray-scale image into a binary image, based on a medial-axis method the skeleton of the pore network is obtained preserving the connectivity, nodes and pore centers on the skeleton are detected, mathematical morphology techniques involving three-dimensional template and dilation are used to obtain the pore and throat geometry data, these geometrical and positional data are stored and the porous media is reconstructed for use in the IPM. The details of the proposed image processing procedure are explained in the Experimental Details section.

### *Segmentation*

In order to obtain a morphological representation of the porous media system and quantify pore-scale parameters a binarization or segmentation process must be implemented to 'divide' the image into two phases: solid and void. Segmentation converts the gray-scale image to a binary image by identifying two populations in the image based on their intensity values. The simplest segmentation techniques employ global thresholding, where the spatial dependency of phases is ignored<sup>73</sup>. One threshold value is used for the entire system, with values below this value assigned to phase one and values above the threshold value assigned to phase two. Usually the threshold value is chosen based on the histogram of the image. This type of algorithm often leads to phase misidentification due to finite resolution effects or noise in real three-dimensional images. Another approach for segmentation is to use local thresholding criteria as described by Oh and Lindquist<sup>74</sup>. This thresholding method utilizes the spatial covariance of the image in conjunction with indicator kriging to determine object edges. Indicator kriging makes the thresholding local and guarantees smoothness in the threshold surface. In this approach, a first pass over the whole image produces population assignments for a fraction of the voxels, in general the nonedge voxels, leaving the remaining fraction unclassified. This segmentation is based on two threshold values,  $t_1$  and  $t_2$ ; intensity values below the lower threshold value are identified as phase one and intensity values larger than the higher threshold value are identified as phase two. The values of  $t_1$  and  $t_2$  are fixed based on a priori knowledge of the intensity histogram of the whole image. Values between the two threshold values,  $t_1$  and  $t_2$ , are assigned to either phase using the maximum likelihood estimate of each phase based on the two-point correlation function, and that completes the segmentation process. Local thresholding is an important aspect of segmentation, particularly for images such as the ones generated by MRI.

## *The Physics of Magnetic Resonance Imaging (MRI)*

### *Introduction*

Magnetic Resonance Imaging (MRI) was first proposed by Lauterbur<sup>75</sup> and Mansfield and Grannell<sup>76</sup>. Since then it has been widely used in the medical field and only recently it is being applied in other scientific and engineering fields to derive quantitative information. Callaghan<sup>77</sup> gives a comprehensive description of the principles and the applications of NMR physics and MRI.

### *Basic NMR Principles*

#### *Nuclear Spin Angular Momentum*

The phenomenon of nuclear magnetic resonance is observed in any nucleus with non-zero nuclear spin. Nuclei having a magnetic moment,  $\mu$ , also possess ‘spin’; a form of angular momentum which is represented by a vector  $\mathbf{P}$ , the magnitude of which according to quantum mechanics is :

$$|P| = \hbar[I(I + 1)] \quad (2)$$

where  $\hbar$  is equal to  $h/2\pi$  ( $h$  is the Planck’s constant) and  $I$  is the spin number. Angular momentum is fully specified by a vector, the direction of which is taken as along the axis of spin such that the spinning body rotates in the sense of a right-handed screw. Quantum mechanics only allows two components of the angular momentum to be specified simultaneously, hence if we know the total value of  $|\mathbf{P}|$ , we can only measure one other component, e.g.  $\mathbf{P}_z$ . For a given quantum number,  $I$ ,  $\mathbf{P}_z$  may take the values:

$$\mathbf{P}_z = \hbar m_I, \quad m_I = I, I-1, I-2, \dots, -I \quad (3)$$

Since, in this work only  $^1\text{H}$  nuclei will be probed the following discussion will be related to spin-1/2 nuclei. A nucleus with  $I = 1/2$  can only take two values of  $m_I$ : +1/2 and -

1/2. In the absence of an external magnetic field, these states are degenerate, i.e. they have the same energy. However, when a magnetic field of strength  $\mathbf{B}_0$  is applied in z-direction the degeneracy is lifted. In terms of the vector model the spins now occupy two discrete energy levels. The spins, of which the z-component of the angular momentum is parallel to the external field, will be in +1/2 energy state. The spins which are anti-parallel to the  $\mathbf{B}_0$  will be at -1/2. The resulting energy of each state will be:

$$U = -\boldsymbol{\mu} \cdot \mathbf{B}_0 = -\mu_z \mathbf{B}_0 \quad (4)$$

The nuclear magnetic moment,  $\boldsymbol{\mu}$ , is related to the nuclear spin angular momentum,  $\mathbf{P}$ , by the gyromagnetic ratio,  $\gamma$  as:

$$\boldsymbol{\mu} = \gamma \mathbf{P} \quad (5)$$

Hence, energy of each state is:

$$U = -\gamma \hbar m_I \mathbf{B}_0 \quad (6)$$

Thus, the energy difference between the two states will be:

$$\Delta U = |\gamma \hbar \mathbf{B}_0| \quad (7)$$

Transition can be induced between the two levels by applying appropriate electromagnetic radiation. The energy of this electromagnetic radiation is equal to  $\hbar \omega_0$ , where  $\omega_0$  is the frequency of radiation, also known as the Larmor frequency. The selection rule governing such transitions is  $\Delta m_I = \pm 1$ . Thus, in an NMR experiment, the frequency,  $\omega_0$ , of electromagnetic radiation required to induce transition between the two nuclear spin energy levels is:

$$\omega_0 = \gamma \mathbf{B}_0 \quad (8)$$

The frequency,  $\omega_0$ , is the resonance frequency of a nucleus with gyromagnetic ratio  $\gamma$  in the presence of a magnetic field  $\mathbf{B}_0$ . Since the gyromagnetic ratio of the  $^1\text{H}$  nucleus is  $26.7519 \times$

$10^{-7} \text{ radT}^{-1}\text{s}^{-1}$  and the magnetic field is typically in the range of 1.5-12T, the resonance frequency is of the order of 70-600 MHz which is in the radio frequency region of the electromagnetic spectrum.

### *NMR Signal*

A macroscopic sample contains an ensemble of many such spins and the NMR signal detected is the vector sum of the individual magnetic moments  $\mu$  of all the nuclei in the sample. The NMR signal is therefore proportional to the population difference of the spin states between which the transitions are occurring. If the populations of the spin states  $+1/2$  and  $-1/2$  of a spin  $1/2$  nucleus are denoted by  $n_\alpha$  and  $n_\beta$ , respectively, the population difference, at thermal equilibrium, between these two states can be derived from Boltzmann's equation to be:

$$n_\alpha - n_\beta = (n_T \Delta U) / (2kT) \quad (9)$$

where  $n_T$  is the total number of spin states ( $n_T = n_\alpha + n_\beta$ ),  $k$  is the Boltzmann's constant and  $T$  is the temperature of the sample.

From this it can be shown that the signal,  $S$ , observed from a system of spins in the presence of an external field  $B_0$  and irradiated by a magnetic field  $B_1$  is given by:

$$S \propto \gamma^4 B_0^2 n_T B_1 g(\omega_0) / T \quad (10)$$

where  $g(\omega_0)$  is the signal shape factor<sup>78</sup>. Since, the NMR signal,  $S$  is directly proportional to the number of spins it provides quantitative information about the system being studied.

Moreover,  $S$  depends heavily on  $\gamma$ ; the value of which for  $^1\text{H}$  is one of the highest of all nuclei. These advantages combined with its 99.99 % natural abundance, makes  $^1\text{H}$  a convenient nucleus to observe using NMR. Higher strength of magnetic field  $B_0$  is also desirable as it enhances the signal magnitude by a power of two.

### *Larmor Precession and Rotating Frame*

In a macroscopic sample containing a large number of spins, the signal observed in any NMR experiment is the resultant magnetization of all the spins. It is convenient to adopt a classical view of the spin system using a vector model. The resultant magnetization is denoted by the vector  $\mathbf{M}$ . The interaction of a magnetic field  $\mathbf{B}$  and  $\mathbf{M}$  gives a torque on the system and is described by the equation,

$$\frac{d}{dt}\mathbf{M} = -\gamma\mathbf{B}\times\mathbf{M} \quad (11)$$

where ‘ $\times$ ’ represents a vector cross product. This corresponds to a precession of the moment about the field at the frequency  $\omega_0 = \gamma B_0$ , the Larmor frequency. In pulse NMR, resonance is achieved by the application of a smaller oscillatory magnetic field,  $B_1$ , perpendicular to the stationary field  $\mathbf{B}_0$  (Figure 2). In a stationary frame of reference, the pulsed field in the  $x$ -direction is equal to  $2B_1\cos\omega t$ . This can be resolved into two components rotating in opposite directions with angular velocities  $\pm\omega$ . Now we define a frame of reference  $x'y'z'$  rotating at  $\omega$ . In the rotating frame, one of the components of  $B_1$  rotates is stationary and is, by convention, taken as in the  $x'$  direction. The second component of  $B_1$  rotates at  $2\omega$  which can be neglected because the frequency is too high to interact with the magnetization. The resultant effective field,  $B_{\text{eff}}$ , in the rotating frame is:

$$\mathbf{B}_{\text{eff}} = (B_0 - \omega/\gamma)\mathbf{k} + B_1 \mathbf{i} \quad (12)$$

where  $\mathbf{k}$  and  $\mathbf{i}$  are the unit vectors along the  $z'$  and  $x'$  axes, respectively.  $B_{\text{eff}}$  is the effective field and precession of the magnetization occurs about this field as described by the equation,

$$\frac{d}{dt}\mathbf{M} = -\gamma\mathbf{B}\times\mathbf{M} \quad (13)$$

At resonance,  $\omega = \omega_0$  and the vertical component of  $B_{\text{eff}}$  vanishes, thus leaving the effective field along the  $x'$  axis.

### *Radio-frequency Pulses and FT-NMR*

In an NMR experiment, the oscillatory field  $B_1$  is provided by a short radio frequency (r.f.) pulse. If the frequency of the r.f. pulse equals that of the Larmor frequency, there is no  $z'$ -component of the effective field and the magnetization precesses as shown in the figure. The initial magnetization  $M_0$  is the equilibrium value in the static field  $B_0$ . After  $B_1$  is applied along the  $x'$ -axis for a time  $t$ , the magnetization rotates through an angle  $\theta = \gamma B_1 t$  and the pulse is called a  $\theta_{\pm x}$  pulse. Hence,  $M_0$  can be rotated into the  $x'$ - $y'$  plane by a  $90^\circ$  pulse and be inverted by application of a  $180^\circ$  pulse.

Given the various kinds of spin interactions the different molecules present in a real system, there exists a range of resonant frequencies  $\Delta\omega$  and, therefore, the 'on-resonance' condition is not strictly applicable simultaneously to all spins within a macroscopic sample. However, as long as  $B_1 \gg \Delta\omega$ ,  $B_{\text{eff}}$  is nearly in the  $x'y'$  plane. The bandwidth of the r.f. pulse of duration  $t_p$  is of the order  $t_p^{-1}$  and a significant turn angle requires that  $\gamma B_1 t_p \sim 1$ . Therefore, the pulse bandwidth is of the order of the Larmor frequency  $\gamma B_1$ . In general, the magnetic field  $B_1$  is fixed in the NMR equipment and the turn angle is adjusted by changing the duration of the r.f. pulse,  $t_p$ . An r.f. pulse with bandwidth larger than the spectral range of a given sample is called a non-selective pulse, or hard pulse. A hard pulse is, thus, capable of exciting all nuclei for a given spin species.

Upon the application of an r.f. pulse, the magnetization  $M_0$  rotates from its equilibrium position. This effectively disturbs the equilibrium distribution of spins between energy levels. The system then returns to equilibrium and the magnetization 'relaxes' back to

its original position. In the laboratory frame, the rotating of the magnetization by the r.f. pulse results in an oscillating component of the magnetization in the x-y plane. This includes and e.m.f. in a receiver coil placed around the sample which is amplified and compared with a reference signal, and then stored as the time domain signal data, which is known as the free induction decay (FID) (figure 3).

Since, the properties of the sample can be readily interpreted in the frequency spectrum, the FID is converted from time-domain to frequency domain. This is achieved by Fourier transformation. The Fourier transformation of a time dependent function,  $f(t)$  can be represented by a frequency dependent function,  $F(\omega)$  as:

$$F(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t} dt \quad (14)$$

In practice, the time domain signal from the NMR experiment is digitized and represented in a finite space of data points. The time interval between pairs of data points is known as the dwell time,  $\Delta t$ . If  $N$  time domain points sampled at a dwell time  $\Delta t$ , the Fourier transformation can be expressed in discrete form as:

$$G(n / N\Delta t) = \sum_{m=-N/2}^{N/2-1} g(m\Delta t)e^{-i2\pi mn / N}, n = -N/2, \dots, N/2 - 1 \quad (15)$$

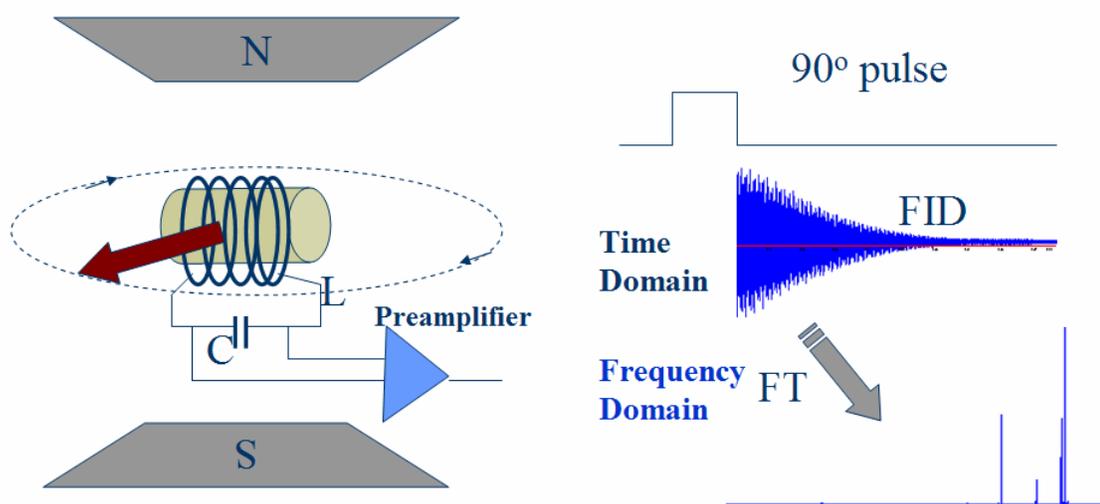
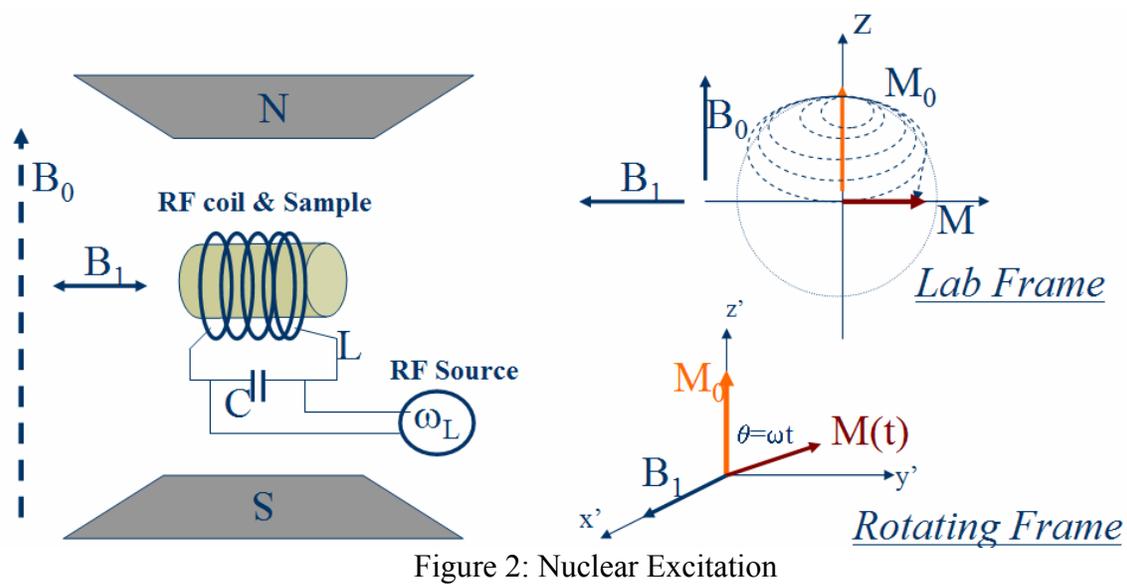


Figure 3: Signal Acquisition and Fourier Transformation

where  $G(n/N\Delta t)$  and  $g(m\Delta t)$  are the discrete forms of  $F(\omega)$  and  $f(t)$ , respectively. The consequence of sampling  $g(mt)$  at intervals of  $t$  is that the transformed function  $G(n/N\Delta t)$  is periodic. When oscillatory data are digitized this way there is a limit to the frequencies which can be distinguished. According to the Nyquist sampling theorem, frequency components with periods less than  $2\Delta t$ , thus  $1/2\Delta t$  is the maximum frequency which can be measured. Another consequence of the discrete transformation is that a spectral component with frequency  $(1/2\Delta t)+\delta$  will appear at  $(1/2\Delta t)-\delta$ . However, this effect can be overcome by quadrature detection, i.e. the components of magnetization which are in-phase and out-of-phase with the r.f. irradiation are both detected simultaneously. The resulting frequency spectrum then has real and imaginary components and phase adjustment can be made after the transformation.

Due to the inherently low sensitivity of NMR, signals from  $n$  successive experiments are summed to enhance the signal-to-noise ratio. In successive addition, the signals add coherently while the noise adds in random phase. The noise amplitude is proportional to  $n^{1/2}$ . The signal-to-noise ratio improves as  $n/n^{1/2}$  or  $n^{1/2}$ . The rate at which successive addition can be performed, in real experiments, is however, limited by the return of the magnetization to its equilibrium value.

### *Relaxation*

After the spins in a sample have been excited by an r.f. pulse, they begin to return to thermodynamic equilibrium by losing the excess energy. This process is known as relaxation and involves the spin interacting (via photon transmission) with a potential energy sink, which can absorb energy at the Larmor frequency. Theories of relaxation can be found in detail in Abragam<sup>79</sup>. Two types of relaxation, spin-lattice or longitudinal relaxation

(characterized by the time constant  $T_1$ ) and spin-spin or transverse relaxation (characterized by the time constants  $T_2$  and  $T_2^*$ ) are of fundamental importance to any NMR experiments.

### *Spin-lattice Relaxation*

At thermodynamic equilibrium, the magnetic moment vector lies parallel to the  $B_0$  field direction i.e. z-direction. Once the magnetic moment has been disturbed from its thermodynamic equilibrium, the spin system contains excess energy. The removal of this energy to the surrounding system is known as spin-lattice relaxation. Spin-lattice relaxation determines the rate at which  $M_z$  returns to its equilibrium state which is parallel to  $B_0$  (Figure 3). In this process, the spin system loses energy to the surrounding lattice. This relaxation can be expressed as an exponential process characterized by the equation,

$$\frac{dM_z}{dt} = -\frac{(M_z - M_0)}{T_1} \quad (16)$$

where  $T_1$  is the spin-lattice relaxation time. The solution of this equation is,

$$M_z(t) = M_z(o)e^{(-t/T_1)} + M_0[1 - e^{(-t/T_1)}] \quad (17)$$

where  $M_z(o)$  is the initial z-component of the magnetization vector and  $M_z(t)$  is the magnetization in the z-direction after a time period of t.  $T_1$  varies greatly with molecular structure and lattice structure. The value of  $T_1$  is determined by the ability of the surrounding lattice to absorb energy at the frequency of the spin precession. Therefore, in solids where the rate of molecular tumbling is very slow,  $T_1$  is generally large, whereas in liquids where the molecules are more mobile and the tumbling is faster,  $T_1$  may be much shorter.  $T_1$  is also reduced by even a small amount of paramagnetic species within the sample.

### *Spin-spin Relaxation*

The decay of transverse magnetization is governed by spin-spin relaxation. The spin-spin relaxation is characterized by a time constant  $T_2$ ,

$$\frac{dM_x}{dt} = -\frac{M_x}{T_2}, \quad \frac{dM_y}{dt} = -\frac{M_y}{T_2} \quad (18)$$

$T_2$  is known as the spin-spin relaxation time or transverse relaxation time. Direct interactions between spins of different nuclei result in relaxation of  $M_x$  and  $M_y$  in addition to other indirect energy exchange via the lattice. Transverse magnetization causes a loss of phase coherence of the spin system in the x-y plane.  $T_2$  relaxation is an irreversible process which does not remove energy from the spin system but the loss of phase coherence causes an increase in entropy. The phase incoherence leads to the exponential decay in the detectable signal, as the superposition of groups of spin in anti-phase negate one another (Figure 4). The relationship between  $T_1$  and  $T_2$  is complex, since spin-lattice and spin-spin processes indirectly share some energy exchange pathways, but it has been shown that always  $T_2 \leq T_1$ <sup>77</sup>. The spin-spin relaxation equation can be solved, with the initial condition of  $M_y(0)=M_0$ , to,

$$M_y(2t_d) = M_0 e^{(-2t_d/T_2)} \quad (19)$$

Several other factors also affect the signal decay characterized by  $T_2$ , especially local heterogeneity in the magnetic field. If we consider two spin isochromats, which are in different regions of the sample and thus experience a different  $\mathbf{B}_0$ , they will have Larmor frequencies which will differ by  $\Delta\omega = \gamma\Delta\mathbf{B}_0$ . A time  $t$  after excitation, these isochromats will have a phase difference of  $\Delta\omega t$  and so there would be a loss of phase coherence associated with this field inhomogeneity. This spin dephasing is reversible and a time constant for this dephasing is defined as  $T_2'$ . The total apparent relaxation,  $T_2^*$  is related

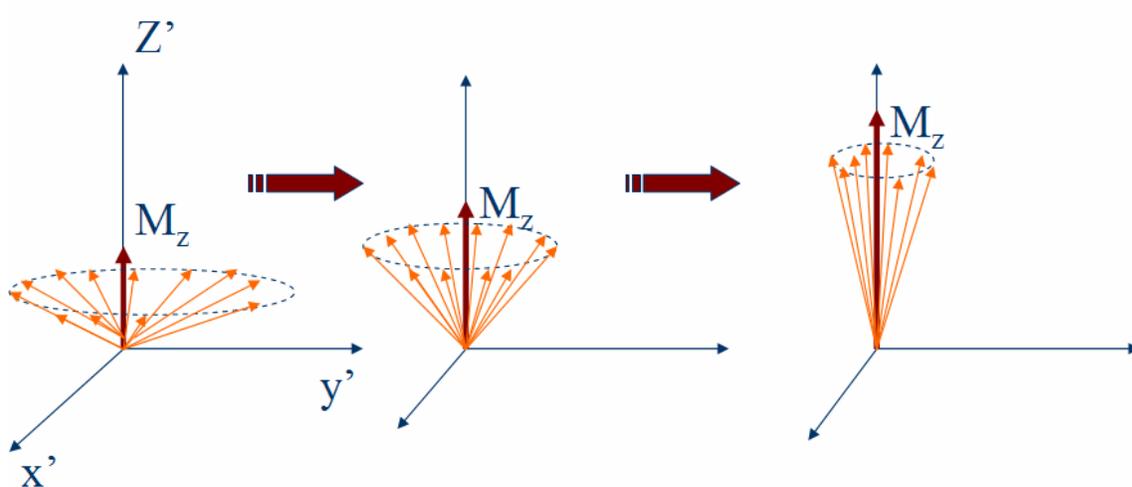


Figure.4: Spin-Lattice or Longitudinal Relaxation Process

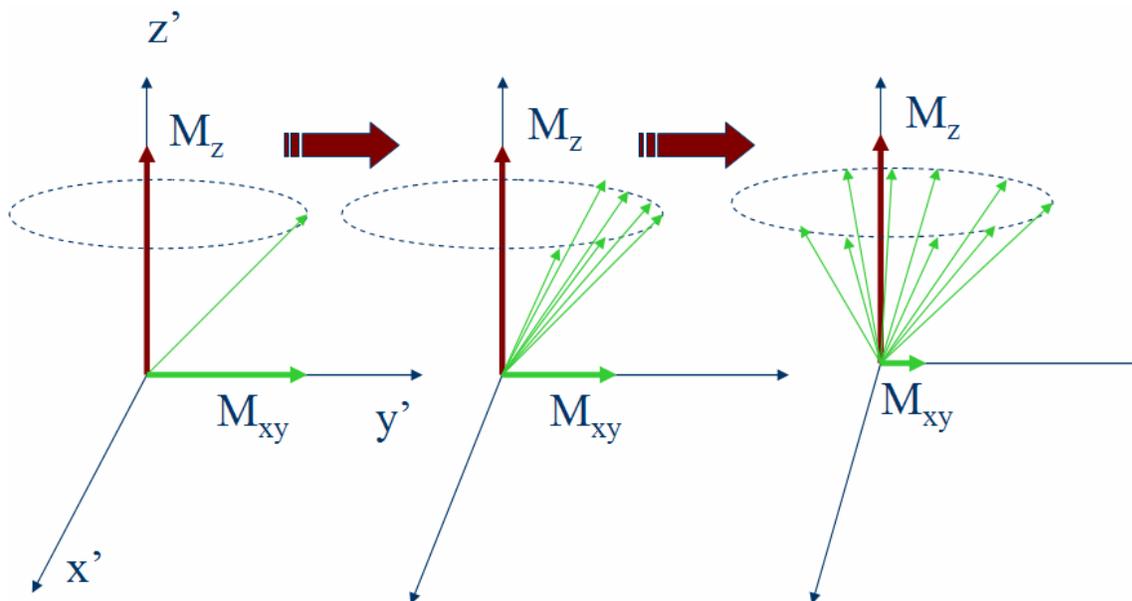


Figure.5: Spin-Spin or Transverse Relaxation Process

to the irreversible spin-spin relaxation constant,  $T_2$ , and the reversible spin dephasing constant,  $T_2'$ , by the expression,

$$\frac{1}{T_2^*} = \frac{1}{T_2'} + \frac{1}{T_2} \quad (20)$$

$T_2^*$  can be obtained from the NMR spectrum of the sample. The effects of  $T_2'$  are not reversed, so the influence of both  $T_2$  and  $T_2'$  processes lead to a decay of the FID, which results in the broadening of the resonance lineshape in the frequency domain. For an exponential decay, which is associated with most single resonant frequency liquids the frequency domain spectrum attained from the Fourier transform of the acquired data produces a Lorentzian lineshape. If the peak width,  $\Delta\nu$ , is taken as the FWHM (full-width at half maximum) then the relaxation constant  $T_2^*$  is given by,

$$T_2^* = 1/(\pi\Delta\nu) \quad (21)$$

Whilst  $T_2^*$  can be determined directly from an FID, the value of  $T_2$ , the irreversible component of  $T_2^*$  is obtained from spin echo experiments.

$T_2^*$  for a pure liquid, in a well shimmed (i.e. homogeneous) magnetic field, may approach the actual  $T_2$  of the liquid with  $\Delta\nu$  below 1Hz. In substances where there are many phase boundaries, for example in porous media, magnetic susceptibility is usually discontinuous at all interfaces between the various phases, resulting in local variations in the magnetic field gradient. Identical nuclei existing in a variety of magnetically non-equivalent sites within the sample will each resonate with different frequencies thereby giving rise to an increase in linewidth, and thus, the presence of multiple phase systems will cause a reduction in  $T_2^*$  when the bulk relaxation rate is measured. Distortions in the static magnetic field due to susceptibility differences can produce a very broad range of Larmor range of Larmor

frequencies. If the spectral width per data point ( $1/\Delta tN$ ) is less than the range of Larmor frequencies some spins will contribute signal to the neighboring voxel and so there will not be full resolution between voxels. To attain the required spatial resolution it is necessary to reduce the dwell or the number of points, both of which will require an increase in the magnetic gradient strength,  $G$ , hence gradients that dominate over the internal gradients set up by susceptibility differences are required.

### ***Image Formation***

In MRI, spatial resolution is achieved by applying a small magnetic field gradient,  $G$ , in addition to the uniform polarizing field  $B_0$ . Therefore, the magnetic field varies linearly with the position of the spins, the angular frequency of the spins is then a function of position in the  $z$ -direction and is given by,

$$\omega_z = \gamma(B_0 + G_z z) \quad (22)$$

where  $G_z$  is the  $z$ -component of the magnetic field gradient  $G$ . Thus, the frequency varies linearly with position. More generally, the spatial distribution of the frequency can be written as,

$$\omega(\mathbf{r}) = \gamma B_0 + \gamma \mathbf{G} \cdot \mathbf{r} \quad (23)$$

where  $\mathbf{r}$  is the position vector of the spin. The transverse magnetization and, therefore, the signal  $dS$  in an element volume,  $dV$ , at position  $\mathbf{r}$  with spin density  $\rho(\mathbf{r})$  is given by,

$$dS(\mathbf{G}, t) = \rho(\mathbf{r}) dV \exp[i(\gamma B_0 + \gamma \mathbf{G} \cdot \mathbf{r})t] \quad (24)$$

This expression makes no allowance for the decay of signal due to transverse magnetization; however, in practice the magnetic field gradient is sufficiently large so that the dephasing of transverse magnetization due to  $\gamma \mathbf{G} \cdot \mathbf{r}$  is much faster than that due to  $T_2$ . In the rotating frame

of reference,  $\gamma B_0$  in equation(24) can be neglected and thus the integrated signal amplitude becomes,

$$S(t) = \iiint \rho(r) \exp[i\gamma G r t] dr \quad (25)$$

It is useful to adopt the definition of the general reciprocal space vector  $k$ <sup>76</sup>,

$$k = (2\pi)^{-1} \gamma G t \quad (26)$$

where  $k$  is a vector quantity possessing three components:  $k_x$ ,  $k_y$  and  $k_z$ . The time dependency of  $k$  and the use of the Fourier transform leads to:

$$S(k) = \iiint \rho(r) \exp[i2\pi k r] dr \quad (27)$$

and

$$\rho(r) = \iiint S(k) \exp[-i2\pi k r] dk \quad (28)$$

From the equation(25), k-space  $[S(k)]$  may be sampled by changing either time or applied gradient magnitude.

### *One-dimensional Spin-echo Imaging*

The simplest imaging experiment is the projection of spin density,  $\rho(r)$ , along an axis. The Hahn spin echo sequence of hard  $90^\circ$  and  $180^\circ$  pulses is applied to produce an echo after a time period of TE from the  $90^\circ$  pulse. A z-gradient  $G_z$  is applied between the two pulses for a time t, causing the spins at position z along the axis of the gradient to acquire a phase shift relative to the static field resonant frequency of  $\gamma G_z z t$ . The refocusing  $180_y$  pulse inverts the phase of all spins, and the second gradient which is twice as long as the first gradient causes the echo at time TE (Figure 6). The spin echo is the resultant of all the spins which are precessing at  $\omega_z$ , given by equation(22). Therefore, a Fourier transform of the acquired signal yields a one-dimensional projection of the spin density along the z-axis.

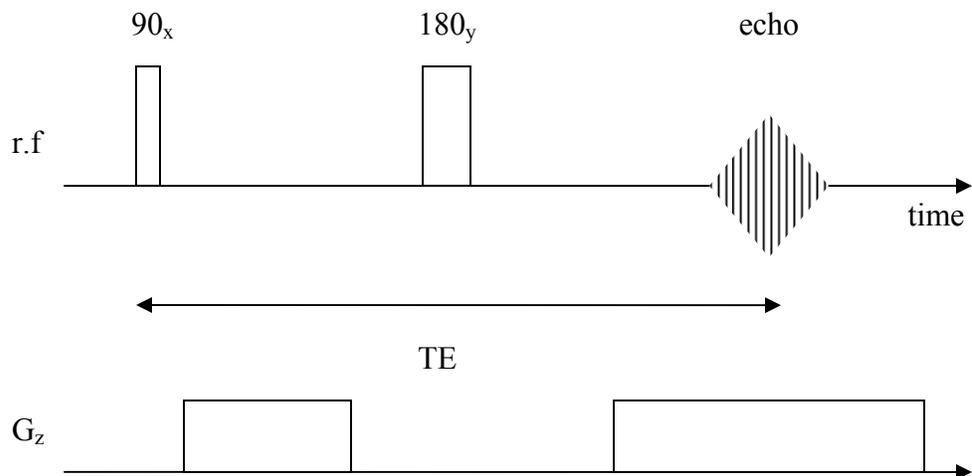


Figure.6: Spin echo pulse sequence used to acquire a one-dimensional signal profile

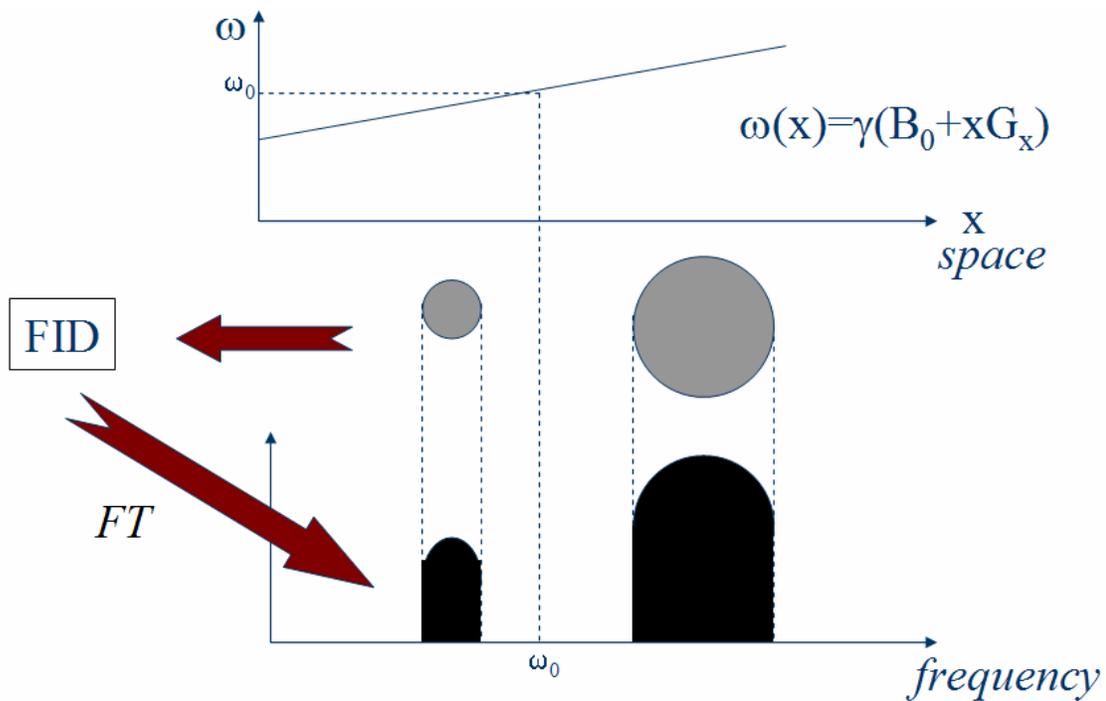


Figure.7: Frequency encoding in one dimensional imaging

### *Selective Excitation and Slice Selection*

Slice selection is achieved by applying a narrow band r.f. field by the use of a soft pulse and gradient field,  $G$ , simultaneously. The resonance frequencies of the spins are along the gradient direction are spread out by the gradient. Therefore, the spins that will be excited are those that resonate within the bandwidth,  $\Delta\omega$ , of the r.f. pulse and consequently only those spins that lie within the slice defined by  $\Delta z$  are observed. The width of the selected slice is controlled by the adjustable parameters  $\Delta\omega$  and  $G_z$ , and thin slice selection is obtained by either narrow-band r.f. excitation or strong gradient  $G_z$  (figure 7). The width of the slice is given by,

$$\Delta z = \Delta\omega / \gamma G_z \quad (29)$$

Hard pulses, which have frequency bandwidths larger than the range of frequencies encountered in the sample, are used when all spins are to be excited. For slice selection, soft pulses are used which have a much smaller frequency bandwidth than hard pulses.

### *Two-dimensional Imaging*

One of the most common techniques used to obtain two-dimensional images is the spin-echo pulse sequence. Many variations of the basic sequence have been developed. The application of the  $180_y$  had pulse results in an echo at time TE after the  $90_x$  pulse. The gradient in the x-direction ( $G_x$ ) applied after the  $90_x$  pulse causes the spins to dephase.  $G_x$  is again applied after the  $180_y$  pulse and a refocussed echo occurs. The second  $G_x$  gradient is adjusted so that the gradient refocused echo coincides with the spin echo. The x-direction described here is usually

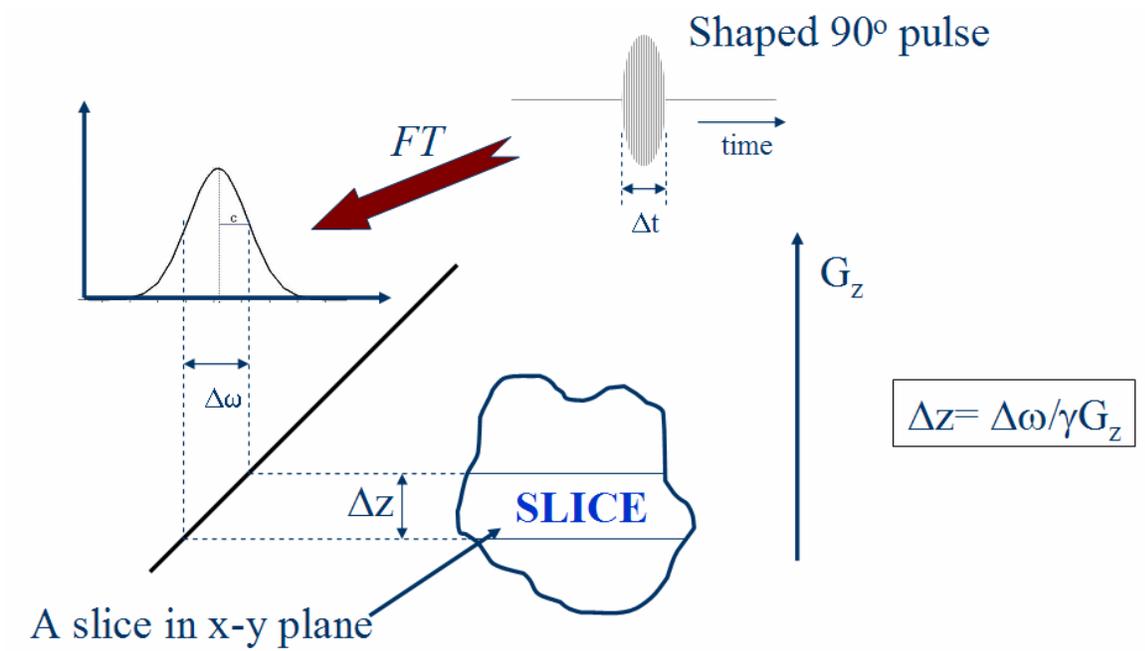


Figure 8: The slice selection process

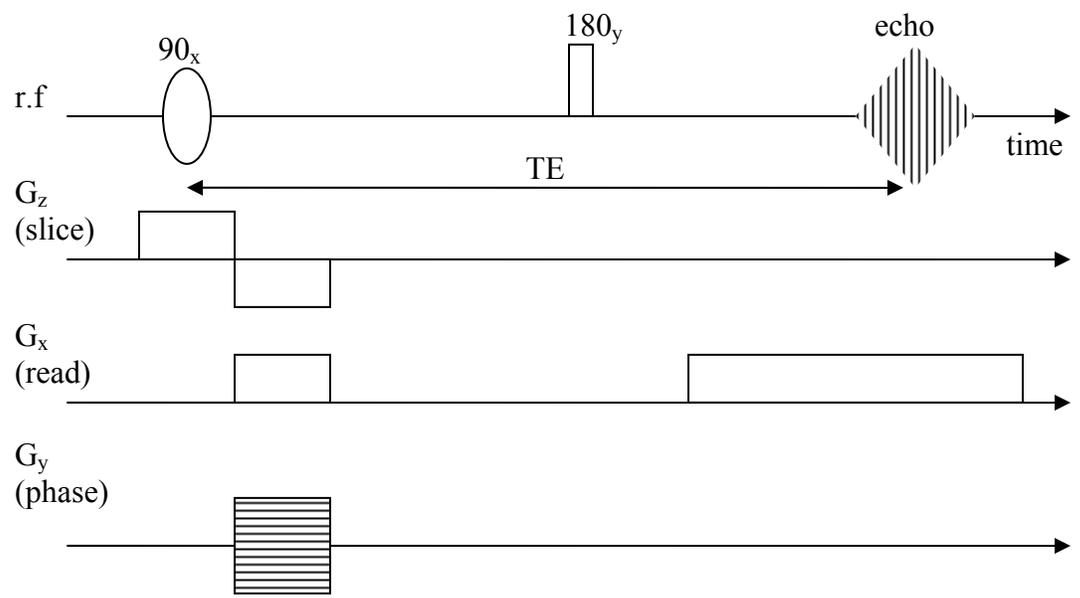


Figure 9: Spin-echo pulse sequence for two dimensional imaging

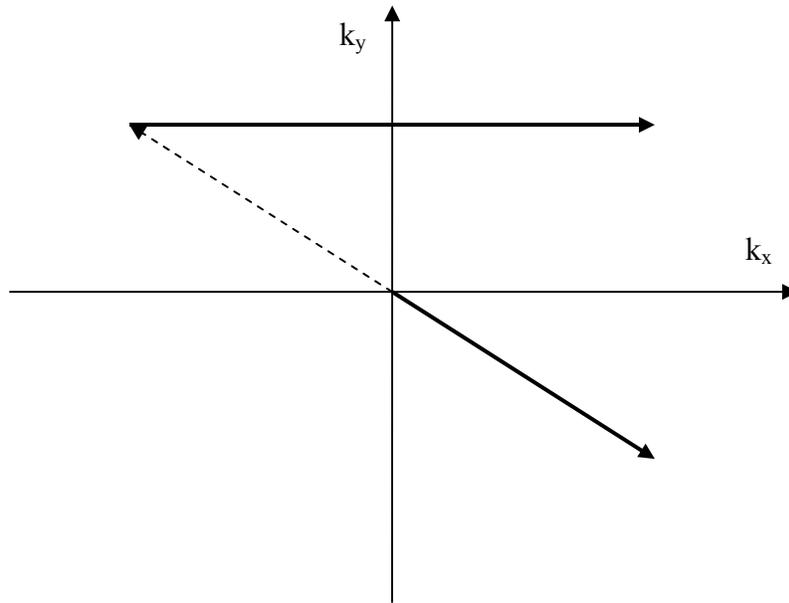


Figure 10: Spin-echo two-dimensional imaging in k-space

called the ‘read’ direction and the x-gradient is named as the ‘read’ gradient. Besides the x- and z-gradients, a y-gradient ( $G_y$ ) is also applied during the first read gradient of a time  $t_y$  and a phase shift  $\phi$  given by,

$$\phi = \gamma G_y y t_y \quad (30)$$

is encoded on the signal. In terms of the k-space notation (Mansfield and Grannell, 1973), this ‘phase’ gradient varies the y-coordinate of the k-space origin, which occurs at the center of the echo. The associated k-space raster of the spin-echo pulse sequence is shown in figure 9. After the selective  $90_x$  pulse, the presence of the ‘read’ and a ‘phase’ gradient moves the k-space origin diagonally across the k-space. The refocusing  $180_y$  pulse then inverts the phase of the spins through the center of k-space and this is represented by the broken arrow on the k-space raster. Finally the k-space is sampled along the read direction upon the application of a constant x-gradient,  $G_x$ . The whole k-space is sampled by increasing the phase gradient in steps while keeping the read gradient constant. This allows the reconstruction of the spin density image according to the equation(28) which can be expressed as,

$$\rho(x,y) = \iint S(x,y) \exp[-i2\pi(k_x x + k_y y)] dx dy \quad (31)$$

If a series of signals are acquired with  $G_y$  incremented over a range of values corresponding to the desired image space, the  $\rho(x,y)$  arising from the two-dimensional Fourier transform becomes an image of the spin density.

Another important technique is the gradient echo imaging sequence. The associated k-space raster is shown in the figure. Instead of using a  $180_y$  pulse, an echo is

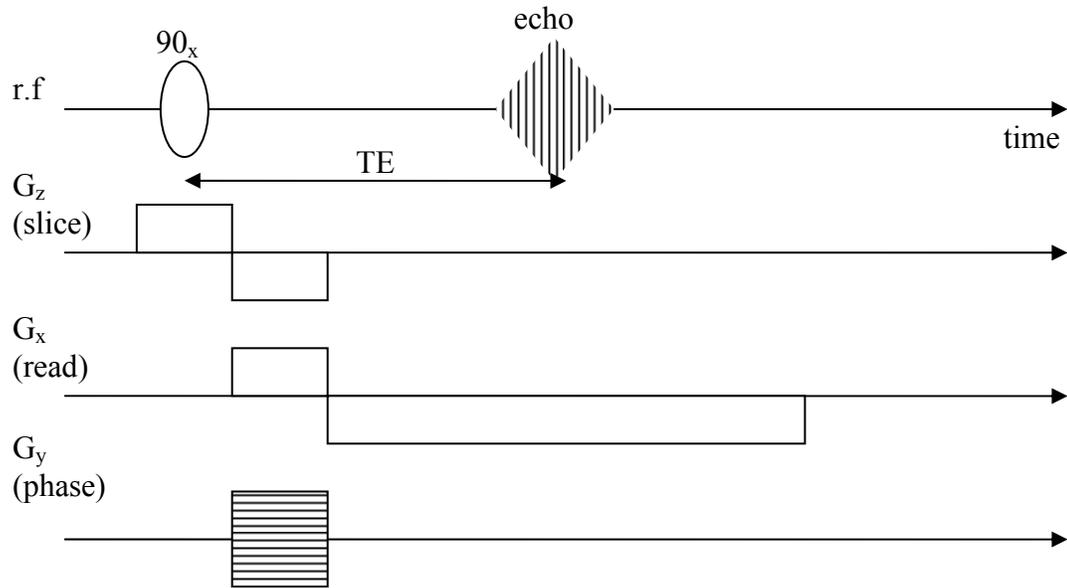


Figure 11: Gradient-echo pulse sequence for two-dimensional imaging

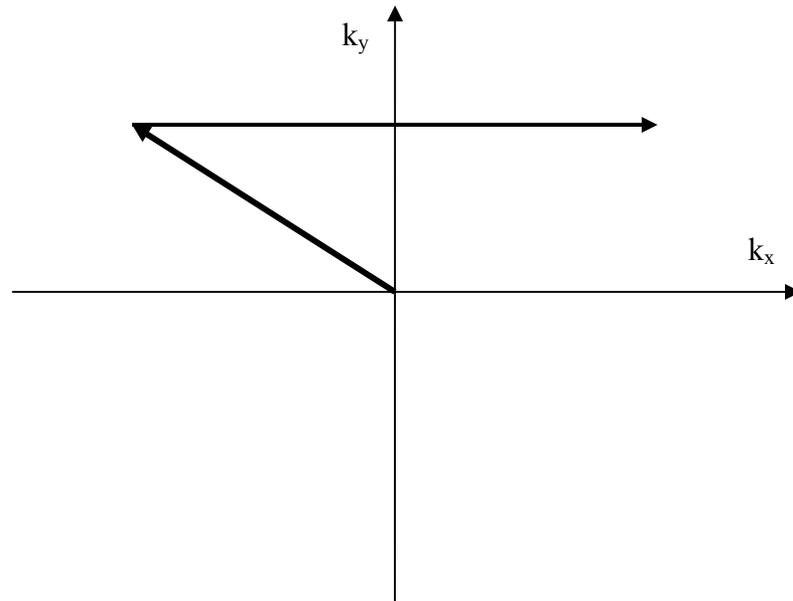


Figure 12: Gradient-echo two-dimensional imaging in  $k$ -space

formed by reversing the polarity of the read gradient. However, due to the absence of the  $180_y$  pulse, gradient reversal only refocuses those phase shifts which are associated with the original gradient applied during the r.f. excitation but not those which are caused by inhomogeneities in the magnetic field. Therefore, the gradient echo technique is subject to  $T_2^*$  relaxation. But, the spin echo technique is capable of refocusing all the phase shifts which are caused by the gradient applied as well as those caused by the inhomogeneities in the magnetic field. The spin echo is thus subject to  $T_2$  relaxation which is slower than the  $T_2^*$  relaxation. However, in case of samples where both  $T_2$  and  $T_2^*$  are very short, the gradient echo technique has advantage over the spin echo technique because of the shorter acquisition time which employs one less  $180_y$  pulse. Haase<sup>80</sup> has extended this technique to allow for very quick image acquisition, where pulse angle of about  $5^\circ$  are employed together with fast repetition times. This technique is also called FLASH (Fast Low-Angle Shot) imaging. However, the signal intensity of the final image is reduced as a result of the low excitation angle and therefore has a lower signal-to-noise ratio compared to the spin echo technique.

### ***X-ray Microtomography***

X-ray microtomography is another non-invasive method to obtain three-dimensional mapping of materials. When x-ray passes through different materials, its attenuation in intensity becomes a function of the material it passed through. Using this contrast in attenuation coefficient of the transmitted x-ray a map of the material is obtained. The x-ray micro computed tomography system obtains multiple x-ray “shadow” transmission images of the object from different angular views, as the object rotates on a high precision stage. From these shadow images, cross-sectional images of the object are reconstructed by a modified Feldkamp cone-beam algorithm<sup>81</sup>, creating a complete three-dimensional representation of

the internal microstructure and density over the selected range of heights in the transmission images. Very high resolution ( $\sim 1\mu\text{m}$ ) three-dimensional images can be obtained using this method, giving an accurate description of the pore-space microstructure. The drawbacks of this method is that it takes long time to acquire one three-dimensional image, and also drying cannot be implemented in-situ since the sample to be imaged has to be put inside a cavity and rotated during imaging. Thus this method is employed to acquire detailed three-dimensional image of randomly packed glass beads to validate the particle packing simulation algorithm.

**References:**

1. Whitaker S., Simultaneous Heat, Mass and Momentum Transfer in Porous Media: A Theory of Drying, *Advances in Heat Transfer*, Vol. 13, Academic Press, New York, 1977, 119–203
2. Masmoudi, W.; Prat, M. and Bories, S. Drying: percolation theory or continuum approach. Some experimental evidences. In *Heat and Mass Transfer in Porous Media* (Edited by Quiantard, M. and Todorovic, M.).817-828. Elsevier, New York.1992.
3. Plumb, O.A. and Prat, M. Microscopic models for the study of drying of capillary porous media. In *Drying '92* (Edited by Mujumdar, A.S.). 397-406. Elsevier, New York.
4. Laurindo J.B. and M. Prat , Numerical and experimental network study of evaporation in capillary porous media. Phase distributions. *Chem. Eng. Sci.* 51 23 (1996), 5171–5185.

5. Laurindo J.B. and M. Prat , Numerical and experimental network study of evaporation in capillary porous media. *Drying rates. Chem. Eng. Sci.* 53 12 (1998), 2257–2269.
6. Prat M. Recent advances in pore-scale models for drying of porous media. *Chemical Engineering Journal*, Volume 86, Issues 1-2, 28 February 2002, Pages 153-164
7. Waananen, K.M., Litchfield, J.B. and Okos, M.R., 1993. Classification of drying models of porous solids, *Drying Technology*, 11,1.
8. J. Bruce Litchfield and Martin R. Okos. Moisture diffusivity in pasta during drying .*Journal of Food Engineering*, Volume 17, Issue 2, 1992, 117-142
9. Michael J. McCarthy' and Ernest Perez et al. Model for Transient Moisture Profiles of a Drying Apple Slab Using the Data Obtained with Magnetic Resonance Imaging. *Biotechnol. Prog.* 1991, 7, 540-543.
10. Ruan, R; Schmidt, A.R.; Litchfield, J.B. Nondestructive measurement of transient moisture profiles and the moisture diffusion coefficients in a potato during drying and adsorption by NMR imaging. *J. Food Process Engineering.* 14:297; 1991.
11. Song, H; Litchfield, J.B. NMR imaging of transient three-dimensional moisture distribution in and ear of corn during drying. *Cereal Chemistry.* 67:580; 1990.
12. B. P. Hills, J. Godward and K. M. Wright. Fast radial NMR microimaging studies of pasta drying. *Journal of Food Engineering*, Volume 33, Issues 3-4, August-September 1997, 321-335.
13. Hollewand, M.P.; Gladden, L.F. Probing the structure of porous pellets : an NMR study of drying. *Magnetic Resonance Imaging.* 12:291-294; 1994.

14. S. D. Beyea, B. J. Balcom, T. W. Bremner, P. J. Prado, D. P. Green, R. L. Armstrong and P. E. Grattan-Bellew. Magnetic Resonance Imaging and Moisture Content Profiles of Drying Concrete. *Cement and Concrete Research*, Volume 28, Issue 3, March 1998, 453-463.
15. Pel, L.; Ketelaars, A.A.J.; Adan, O.C.G. et al. Determination of moisture diffusivity in porous media using scanning neutron radiography. *Intl J of Heat and Mass Trans.* 36:1261–1267;1993.
16. Koptuyug, I.V.; Kabanikhin, S.I.; Iskakov, K.T., et al. A quantitative NMR imaging study of mass transport in porous solids during drying. *Chemical Engineering Science.* 55:1559-1571; 2000.
17. F. de Barquin and J. M. Dereppe. Drying of a white porous limestone monitored by NMR imaging. *Magnetic Resonance Imaging*, Volume 14, Issues 7-8, 1996, 941-943.
18. Schrader, G.W.; Litchfield, J.B. Moisture profiles in a model food gel during drying : Measurement using magnetic resonance imaging and evaluation of the Fickian diffusion model. *ASAE Proceedings*, Albuquerque, New Mexico, Paper No. 91-6055; 1991.
19. M.A. Ruiz-Cabrera, L. Foucat, J.M. Bonny, J.P. Renou and J.D. Daudin. Assessment of water diffusivity in gelatine gel from moisture profiles. I—Non-destructive measurement of 1D moisture profiles during drying from 2D nuclear magnetic resonance images. *Jurnal of Food Engineering*, Volume 68, Issue 2, May 2005, 209-219.
20. Gilliland and. TK. Sherwood, Drying of. Solids; Diffusion Equations for Period of Constant Drying Rate. *Ind. Eng. Chem.* 25. 1134-36 (1933).

21. Dietl C, George OP, Bansal NK. Modeling of diffusion in capillary porous materials during the drying process. *Drying Technology*.1995;13:267-79.
22. Rogers JA, Kaviany M. Funicular and evaporative front regimes in convective drying of granular beds. *Int. J. Heat mass Transfer*. 1992;35: 469-82.
23. Chen P, Schmidt PS. An integral model for convective drying of hygroscopic materials. In: Mujumdar AS, Roques MA. *Drying '89*. Hemisphere, Washington, DC. 1989:162-69.
24. Boukadida N, Ben Nasrallah S. Two Dimensional Heat and Mass Transfer During Convective Drying of Porous Media. *Drying Technology*. 1995;13:661-695.
25. Whitaker, *Coupled Transport in Multiphase Systems: A Theory of Drying*, *Advances in Heat Transfer*, Vol. 31, Academic Press, New York, 1998, 1–103.
26. Comings EW, Sherwood TK. The drying of solids VII: moisture movement by capillarity in drying of granular materials. *Ind. Eng. Chem*. 1934;26:1096-1105.
27. Kisakurek B, Gebizlioglu O. *Proceedings of the first international symposium on drying*. 1978:56-59.
28. Gunstensen AK, Rothman DH. Lattice Boltzman studies of immiscible two-phase flow through porous media. *Journal of Geophysical Reviews*. 1993;98:6431–41.
29. Rothman DH, Zaleski S. Lattice-gas models of phase separation: interfaces, phase transitions, and multiphase flow. *Reviews of Modern Physics*. 1994;66:1417–79.
30. Dawson SP, Chen SY, Doolen GD, Janecky DR, Lawniczak A. Lattice methods and their applications to reacting systems. *Computers and Chemical Engineering*. 1995;19:617–46.

31. Blunt M, King P. Macroscopic parameters from simulations of pore scale flow. *Physical Review A*. 1990;42:4780–87.
32. Rajaram H, Ferrand LA, Celia MA. Prediction of relative permeabilities for unconsolidated soils using pore-scale network models. *Water Resources Research*. 1997;33:43–52.
33. Fischer U, Celia M. Prediction of relative and absolute permeabilities for gas and water retention curves using a pore-scale network model. *Water Resources Research*. 1999;35:1089–1100.
34. Jerauld GR, Salter SJ. The effect of pore-structure on hysteresis in relative permeability and capillary pressure: pore level modeling. *Transport in Porous Media*. 1990;5:103–51.
35. Bryant SP, King R, Mellor DW. Network model evaluation of permeability and spatial correlation in a real random sphere packing. *Transport in Porous Media*. 1993;11:53–70.
36. Reeves PC, Celia MA. A functional relationship between capillary pressure, saturation, and interfacial areas as revealed by a pore-scale network model. *Water Resources Research*. 1996;32:2345–58.
37. Thompson KE, Fogler HS. Modeling flow in disordered packed beds from pore-scale fluid mechanics. *AIChE Journal*. 1997;43:1377–89.
38. Lowry MI, Miller CT. Pore-scale modeling of nonwetting-phase residual in porous media. *Water Resources Research*. 1995;31:455–73.
39. Dillard LA, Blunt M. Development of a pore network simulation model to study nonaqueous phase liquid dissolution. *Water Resources Research*. 2000;36:439–54.

40. Dias MM, Payatakes AC. Network models for two-phase flow in porous media: motion of oil ganglia. *Journal of Fluid Mechanics*. 1986;164:337–68.
41. Li Y, Wardlaw NC. Mechanisms of nonwetting phase trapping during imbibition at slow rates. *Journal of Colloid and Interface Science*. 1986;109:461–72.
42. Li Y, Wardlaw NC. The influence of wettability and critical pore-throat size ratio on snap-off. *Journal of Colloid and Interface Science*. 1986;109:473–86.
43. Satik C, Yortsos YC. A pore network study of bubble growth in porous media driven by heat transfer. *J. Heat Transf. Trans. ASME*. 1996;118:455–62.
44. Li X, Yortsos YC. Visualization and simulation of bubble growth in pores networks. *AIChE J*. 1995;41:214-22.
45. McCall KR, Guyer RA. Fluid configurations in partially saturated porous media. *Phys. Rev. B*. 1991;43:808-15.
46. Prat M. Percolation model of drying under isothermal conditions in porous media. *Int. J. Mult. Flow*. 1993;19:691–704.
47. Prat M. Isothermal drying of non-hygroscopic capillary-porous materials as an invasion percolation process. *Int. J. Mult. Flow*. 1995;21:875–92.
48. Le Bray Y, Prat M. Three-dimensional pore network simulation of drying in capillary porous media. *Int. J. of Heat and Mass Trans.* 1999;42: 4207-24.
49. Broadbent SR, Hammersley JM. The mathematics of percolation in networks. *Proc. Cambridge Philosophical Society*. 1957;53:629-72.
50. Chandler R, Koplik J, Larman K, Willemsen J. Capillary displacement and percolation in porous media. *J. Fluid Mech*. 1982;119:249-58.

51. Wilkinson D, Willemsen JF. Invasion percolation: a new form of percolation theory. *J. Phys. A*. 1983;16:3365–76.
52. Wilkinson D. Percolation model of immiscible displacement in the presence of buoyancy forces. *Phys. Rev. A*. 1984;30:520–31.
53. Glass RJ, Yarrington L. Simulation of gravity fingering in porous media using a modified invasion percolation model. *Geoderma*. 1996;70:231-47.
54. Shaw TM. Drying as an immiscible displacement process with fluid counterflow. *Phys. Rev. Lett*. 1987;59:1671–74.
55. Celia MA, Reeves PC, Ferrand LA. Recent advances in pore scale models for multiphase flow in porous media. *Reviews of Geophysics*. 1995;33:1049–57.
56. D'Hollander EH. Estimation of the pore size distribution from the moisture characteristic. *Water Resources Research*. 1979;15:107–12.
57. Mishra BK, Sharma MM. Measurement of pore-size distributions from capillary pressure curves. *AIChE Journal*. 1988;34:684–87.
58. Liang ZR, Phillippi PC, Fernandes CP, Magnani FS. Prediction of permeability from the skeleton of three-dimensional pore structure. *SPE Reservoir Evaluation and Engineering*. 1999;2::161–68.
59. Vogel HJ. A numerical experiment on pore size, pore connectivity, water retention, permeability and solute transport using network models. *European Journal of Soil Science*. 2000;51:99–105.
60. Vogel HJ, Roth K. A new approach for determining effective soil hydraulic functions. *European Journal of Soil Science*. 1997;49:547–56.

61. Vogel HJ, Roth K. Quantitative morphology and network representation of soil pore structure. *Advances in Water Resources*. 2001;24:233–42.
62. Ioannidis MA, Chatzis I. On the geometry and topology of 3D stochastic porous media. *Journal of Colloid and Interface Science*. 2000;229:323–34.
63. Hidajat I, Rastogi A, Singh M, Mohanty KK. Transport properties of porous media from thin-sections. SPE 69623, SPE LACPEC. 2001. March 25–27.
64. Spanne P, Thovert JF, Jacquin CJ, Lindquist BW, Jones KW, Adler PM. Synchrotron computed microtomography of porous media: topology and transports. *Physical Review Letters*. 1994;73:2001–04.
65. Hazlett RD. Simulation of capillary-dominated displacements in microtomographic images of reservoir rocks. *Transport in Porous Media*. 1995;20: 21–35.
66. Coker DA, Torquato S. Morphology and physical properties of Fontainebleau sandstone via a tomographic analysis. *Journal of Geophysical Research*. 1996;101:17497–506.
67. Rintoul MD, Torquato S, Yeong C, Kaene DT, Erramilli S, Jun YN, Dabbs DM, Aksay IA. Structure and transport properties of a porous magnetic gel via X-ray microtomography. *Physical Review E*. 1996;54:2663–69.
68. Baldwin CA, Sederman AJ, Mantle MD. Determination and characterization of the structure of a pore space from 3D volume images. *J. of Colloid and Interface Science*. 1996;181:79-92.
69. Lindquist WB, Venkatarangan A. Investigating 3D geometry of porous media from high resolution images. *Physics and Chemistry of the Earth (A)*. 1999;25:593–99.

70. Wildenschild D, Hopmans JW, Vaz CMP, Rivers ML, Rikard D. Using x-ray computed tomography in hydrology: systems, resolutions, and limitations. *J. Hydrology*. 2002;267:285–97.
71. Hazlett RD. Simulation of capillary-dominated displacements in microtomographic images of reservoir rocks. *Transport in Porous Media*. 1995;20:21–35.
72. Zhao HQ, Macdonald IF, Kwiecien MJ. Multi-orientation scanning: a necessity in the identification of pore necks in porous media by 3-D computer reconstruction from serial section data. *Journal of Colloid and Interface Science*. 1994;162:390–401.
73. Fisher, R., Wolfart, E., Perkins, S. and Walker, A. *Hypermedia Image Processing Reference (HIPR)*. 1997. Wiley, New York.
74. Oh W, Lindquist WB. Image thresholding by indicator Kriging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1999;21:590–601.
75. Lauterbur PC. Image formation by induced local interactions: examples employing nuclear magnetic resonance. *Nature*. 1973;242:190-191.
76. Mansfield P, Grannell PK. NMR 'diffraction' in solids? *J. Phys. C: Solid State Phys.* 1973;6:L422-L426.
77. Callaghan PT. *Principles of Nuclear Magnetic Resonance Microscopy*. 1991. Clarendon Press, Oxford, UK.
78. Harris RK. *Nuclear magnetic resonance spectroscopy: a physiochemical view*. 1986. Longman Publishing Group, NY.
79. Abragam A. *Principles of Nuclear Magnetism*. 1983. Oxford University Press, NY.
80. Haase A. Principles and applications of FLASH NMR imaging. *Magnetic Resonance Materials in Physics, Biology and Medicine*. 1994;2: 157-60.

81. Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *Journal of the Optical Soc. of America A*. 1984;1:612-20.

## CHAPTER III

## TITLE

Simulation of Unconsolidated Porous Media and Pore-Space Geometry Characterization

## AUTHORS

Supriyo Ghosh, Yong Pan, Michael Bills, Kevin M. Keener, Paula Chmielewski

## KEY WORDS

Porous media, reconstruction, mathematical morphology, pore-network

## ABSTRACT

Porous media are heterogeneous systems. The microstructures of the pore spaces influence their transport properties. A quantitative geometrical characterization of the pore space is crucial for accurate prediction of porous media transport. Thus, a three-dimensional simulation of porous media was developed based on randomly packed glass beads.

Unconsolidated porous media were reconstructed through Monte Carlo gravitational particle packing simulation. A mathematical morphology based three-dimensional image processing algorithm was developed to characterize the pore space in the simulated porous media. This algorithm calculated the bulk porosity and average particle contact numbers of the porous media. It also generated the pore-throat network with details of pore size distribution, location, and throat tortuosity distribution. The simulation results were validated by statistical comparison of the bulk porosity and pore size distribution obtained from x-ray microtomographic images of randomly packed glass beads. The pore network geometry was successfully extracted from the three-dimensional pore-space. The pore-network characteristics of simulated and experimental beds were found to be statistically equivalent with the level of significance ( $\alpha$ ) as 0.05. Simulation of porous media, given a specific size

distribution of constituent particles, followed by pore-space characterization provided a powerful tool for doing virtual experiments and predicting transport processes of porous material based products.

## INTRODUCTION

Porous media structure consists mainly of two phases, the grains and the pore-space. Most of the porous media are heterogeneous systems with respect to pore-space topology; if the pore-space geometry is studied in detail, a wide spatial variation is observed. Fluid transport through porous materials is encountered in many industrial processes, and is strongly influenced by the pore-space topology. This pore-space geometry is in turn also influenced by the properties of the constituent particles forming the porous material. It is often challenging to mathematically describe porous media because of the disordered nature of its microstructure. Hence transport behavior of porous media is usually correlated with bulk structural properties like porosity/density, pore size distribution, etc. Many efforts have been made to develop continuum mechanics based models to describe porous media transport.<sup>1,2</sup> These mathematical continuum models often correlate the transport behavior of a specific porous medium with their bulk porosity.<sup>3</sup> Sometimes, in these models, the liquid permeability of the porous medium is represented as a function of porosity as well as an average tortuosity, in order to explain the experimental data.<sup>4</sup> These descriptions of porous media flow are not purely mechanistic models, but based on semi-empirical relationships among capillary pressure, saturation level, and relative permeability, obtained through controlled experiments.<sup>5</sup> These models do not account for the detailed microstructure in the porous medium. But, the transport of fluids and solutes in porous media is a strong function

of its geometry and topological characteristics. The microscale phenomena happening in the pore-scale level translates very well into the macro-scale behavior of a porous structure.<sup>6</sup> Hence, pore-network based modeling of various properties of porous media is a powerful method to generate realistic prediction of macroscopic behavior. Early attempts at pore network modeling describe porous media transport based on capillary tubes.<sup>7</sup> These models attempted to explain the permeability without accounting for the interconnectivity among the pore channels. More recently the concept of 2D and 3D networks to describe the pore-space topography has been introduced.<sup>8,9</sup> 2D pore networks have limited application since most porous media involve 3D flow and hence the connectivity between pores can not usually be adequately defined in 2D.<sup>10</sup> The existing 3D network models in the literature are based on networks built using certain assumptions regarding the pore and throat size distribution, and the pores are assumed to be connected through straight cylindrical throats.<sup>8</sup> This does not completely reflect the pore-space geometry in real porous media.

Thus, there is a need to develop a realistic 3D pore network model, which shows more promise to a realistic prediction of porous media transport. Pore network models can be generated using indirect or direct methods. In indirect method, an equivalent network is produced based on distributions of major pore-space structures, pore body and pore throat, and their positional correlations.<sup>11</sup> In contrast, the direct method, extracts the pore-throat network from the pore space directly using 3D image. The direct method requires no assumption related to the topological positions or dimensions. In order to directly map the pore network, a 3D data representing the pore-space, with enough resolution, is essential. Many attempts have been made to generate this dataset using high resolution non-invasive three-dimensional imaging techniques like laser scanning confocal microscopy<sup>12</sup>, x-ray

microtomography<sup>13</sup>, magnetic resonance imaging<sup>14</sup>, etc. There are three major limitations of this approach:

- one is restricted to investigating only those porous media, of which one already has samples of. Thus, one can not use this approach for virtual product design.
- the pore-space data set resolution is limited by the instrument capability that is used to acquire the 3D data.
- the contrast in the dataset. The edges of the pore-space, i.e. the grain boundary in the instrument-generated images will not be absolute. This sharpness of the edges will depend on how the sample material interferes with the instrument signal. The imperfection in obtaining the correct edges can introduce serious errors in the pore network extracted.<sup>15</sup>

These major limitations are avoided in this work by extracting the pore network from a digitally generated consolidated porous media through computer simulation and mathematical morphology based image processing.

A random packing of particles represents an unconsolidated porous media. Various methods have been used by different researchers to generate a random packing of particles. Some of them are: sequential addition<sup>16</sup>, ballistic drop with mechanical interaction calculation<sup>17</sup>, region growing<sup>18</sup>, and mechanical contraction<sup>19</sup>. Simple rules of sequential addition, where particles are added at random positions as long as they do not share spaces with each other, can generate random particle packs, but the packing tend to be fairly loose.<sup>20</sup> The reason for this is while placing a new particle it stops moving as soon as it hits another and do not slide along other particles boundary, which is more realistic. Ballistic drop method of random particle packing simulation utilizes calculations of mechanical forces each time a falling particle encounters another on its path. This method accurately follows the

mechanism of the process, but detailed calculations in each step of the movement are needed. Hence, when a large number of particles are involved in the simulation, use of significant computing power becomes essential, because of the parallel processing nature of the process. Thus, this method of simulation tends to be slower, and needs considerable amount of processor time to generate a random packing with fairly large numbers of particles. The region growing method of random particle packing is used in simulating aggregated powder particles. It starts with one central particle and attaches particles of random size at random available locations. This method is not representative of the process of randomly dropping particles into a container to generate an unconsolidated porous media.<sup>18</sup> The mechanical contraction method of random particle packing is motivated by simulation of amorphous packing and is based on the idea of density quenching a system, which undergoes no thermal fluctuations. Three-dimensional Monte Carlo simulation under gravity is a method of random particle packing, which closely resembles random gravitational dropping of particles and at the same time computationally less demanding. Hence this method was selected to generate the simulated unconsolidated porous media in this work. This method allowed the reconstruction of porous media from any given size distribution of constituent particles.

Since the porous media models are constructed through computer simulations, the resolution of the pore-space was limited only by the available memory (RAM) and processor speed of the computer used. This model enabled perfect binarization of 3D pore-space images without any error from edge or boundary detection.

A 3D image processing algorithm was developed for this work to quantify the detailed pore-space geometry. Skeletonization of the 3D pore-space was the first step in mapping of the pore-throat network. This conversion of the 3D image into the pore-throat

network has a wide variety of applications and has been an active area of research for many decades.<sup>21</sup> Many researchers have tried different sequential, parallel, as well as non-iterative thinning techniques towards the same goal; Hamilton-Jacobi<sup>22</sup>, local flux driven extraction<sup>23</sup>, three-dimensional template based exclusion rule<sup>24</sup>, medial axis extraction<sup>25</sup>, are to name a few. These algorithms, in spite of being effective in finding the skeleton in specific kind of structural features, fail to extract correct centerlines in other types of shape features.<sup>26,27,28</sup> When the sample is as complex pore-space structure, it becomes very challenging to come up with a single algorithm that will converge to a perfect skeleton in an unsupervised (without human intervention during the extraction process) manner. A morphological thinning algorithm was developed in this work which thinned the pore-space with same flux vector from all directions while preserving the continuity. This algorithm also generated the bulk porosity and the co-ordination number for constituent particles. Using the distance-transform principle; pore size distribution, and average tortuosity of the pore-space were also calculated in an automated manner by this algorithm. All these values directly calculated from the three-dimensional morphology of the pore-space were more realistic input parameters for pore network based modeling as compared to networks built based on certain assumptions regarding the pore size distribution.

## METHODS

### *Reconstruction Simulation*

The three-dimensional Monte Carlo simulation algorithm in this work tries to simulate simultaneous random dropping of spherical particles, of a specific size distribution, into a container. As these particles finally take up mutually most stable positions, we obtain

an unconsolidated porous media. The algorithm is built upon six assumptions, which are consistent with the supporting experimental data:

1. the particles are spherical in shape
2. particle radii belong to a user specified distribution
3. particles are rigid
4. the container walls are rigid
5. particles cannot move against the gravity i.e. upwards during their random motion
6. downward and sidewise movement vectors of the particles are random.

The algorithm starts with assignment of the particle number, its radius distribution shape, mean and width. It also prompts the user to provide the dimension of the container. Initially all the particles are allotted their center positions uniformly distributed in the container inner space. No location is approved if it does not obey the assumptions three and four, and that particle is reassigned another permissible location. During each iteration of the simulation, the particles are moved sidewise and downward. The movement direction and magnitude are obtained by a normally distributed random number generation between -1 and +1 (in case of z-direction movement between -1 and 0), followed by multiplication with a factor commensurate with the length-scale of the system. As each iteration is completed, the total potential energy of the particle system with respect to the bottom floor of the container is calculated and stored in an array. With progress of the iterations the particle assembly gradually settles down towards the bottom of the container in a random fashion. The total potential energy of the system drops faster during initial few iterations and then gradually the rate decreases. As the slope of this total potential time-course becomes very close to zero, the iteration stops, and the final location of the particle centers and corresponding radii are

recorded in proper matrices. A flow diagram of the algorithm is given in figure 1. An example of ‘movement of particles’ during a Monte Carlo packing simulation is shown in figure 2. The time-course of the corresponding normalized total potential energy of the system as the simulation progresses is given in figure 3. A three-dimensional matrix is created based on the particle location and radii information. In this matrix the voxels occupied by the particles are assigned 0, and its complement set is assigned 1. This matrix now represents the pore-space. By adding all the voxels belonging to the pore space followed by division with the total number of voxels in the container gave the bulk porosity of the reconstructed porous medium.

#### *Particle Packing Experimentation*

In order to validate the simulation algorithm, packed beds of spherical glass beads were obtained and three-dimensional images were acquired using x-ray microtomography. X-ray micro-CT 3D imaging of 250 $\mu$ , 500 $\mu$  diameter bead packs were obtained using micro-CT 40 model of x-ray imager (Scanco Medical, Zurich, Switzerland). The following acquisition parameters were used in the imaging experiment: X-ray tube current of 57  $\mu$ A, peak energy of 70 kVp, integration time of 250 msec, 1000 projections. During this measurement, x-ray passed through the glass bead pack. The attenuation in intensity of the x-ray as it passes through glass and air are distinctly different. Utilizing this contrast in attenuation coefficient of the transmitted x-ray, a map of the glass bead pack was obtained. The x-ray microtomography system obtained multiple x-ray “shadow” transmission images of the packed bed from different angular views, as it rotates on a high precision stage. From these shadow images, cross-sectional images of the glass bead pack were reconstructed by a modified Feldkamp cone-beam algorithm<sup>29</sup>, finally creating a complete three-dimensional

representation of the internal microstructure. This image data were segmented into pore space and grains based on intensity difference, and the bulk porosities were calculated. The glass beads from the same lot were spread over a light microscope stage, in small batches, with bottom light configuration, and two-dimensional images were acquired. Images were obtained for approximately 1000 beads. An example image is shown in figure 4. An algorithm was developed to automate the process of finding the beads in those images, calculating their radii, and storing them in a data array. This array was then used to determine the shape, mean, and width of the distribution of the bead sizes. The distribution obtained for 250 $\mu$ , 500 $\mu$  diameter beads are shown in figure 5 and 6. This algorithm was validated by manually measuring 500 beads under the same microscope. These distributions were used as one of the input parameters while running the Monte Carlo packing simulation in triplicate. In the experiment, only spherical hard glass beads were used and they were deposited inside a hard plastic cuvette of parallelepiped shape to create the unconsolidated porous media. Hence, the assumptions used the 3D Monte Carlo particle simulation were consistent with the experiment. The bulk porosity, and pore size distributions were then obtained from the simulated packing and x-ray microtomography data, and compared statistically.

### *Image Processing*

The image processing algorithm starts with the pore-space binarized three-dimensional data set. Each iteration in the algorithm executed the following set of actions.

(i) The outer-most thinnest surface of the pore-space was determined. This was accomplished by finding those voxels which belong to the pore space but have at least one grain voxel in its 26-neighborhood (six faces, twelve edges, and eight corners). This set of voxel was stored in

another matrix; their elements were summed, and divided by the total number of voxels belonging to the porous medium in order to calculate the specific surface area of the system.

- (ii) Each of these voxels were taken one by one, eliminated and checked in their 5 voxel x 5 voxel x 5 voxel vicinity if a new three-dimensional object was created because of this elimination. If the outcome was negative then the voxel is designated eligible for elimination.
- (iii) All the eligible voxels in the current iteration were removed from the pore-space data set.
- (iv) The resulting thinned three-dimensional data set was stored in a buffer matrix and the original matrix became available to the next iteration.
- (v) The current buffer matrix was compared for voxel-to-voxel correlation with the previous one. If the outcome was negative then the algorithm continued to the next iteration step.

As a result of following these rules the algorithm stopped on its own as it generated the single voxel thin connected skeleton of the pore-space. A flow chart explaining the skeletonization algorithm is given in figure 7. Next, a Euclidean distance transform was done on the complement of the original pore-space matrix, and this three-dimensional map was multiplied voxel by voxel to the binary skeleton matrix. On the skeleton matrix, the voxels were found which had three or more neighbors which belonged to the same skeleton. These voxels were designated as the pore centers. The distance-transform values of these voxels corresponded to the radius of the pore, which it was the center of. These calculated values of the radii were stored in an array, which translated into the pore size distribution of the reconstructed porous medium. A spherical template based generation of the pores was done based on their respective maximal possible radii. The minimum of the distance-transform map values of all the voxels, belonging to the part of the skeleton that joins two of these pore centers, was the critical radius of that particular throat. The total number of voxels making

the current throat was divided by the linear distance between two connected pore centers and was saved as the tortuosity of the corresponding throat. It is illustrated in figure 8.

In order to validate this algorithm, first, a cubic lattice was formed. One sphere was placed at each corner of the cubes. The radii of the spheres were made equal to one half of the sides of the cubes. The arrangement of the spheres is shown in figure 9(a). Now the pore-space not occupied by these spheres was obtained, and passed on to the skeletonization algorithm. The skeleton thus generated is shown in figure 9(b). The pores generated by the algorithm had all equal radii of the magnitude of 0.83 times the radii of the spheres, which corresponded very well with the analytically calculated value of 0.8284. The skeleton generated in this case also corresponded well with the expected map of it i.e. another cubic lattice.

#### *Statistical Method*

Kolmogorov-Smirnov statistic is used in this work to quantitatively compare two distributions, and determine the statistical equivalence between them. This a standard statistical method of comparing two distributions when the distributions cannot be represented by a well-defined formula like normal distribution, lognormal distribution, etc. Kolmogorov-Smirnov test statistic is the maximum of absolute differences between two cumulative distribution functions of the distributions being compared<sup>30</sup>.

## RESULTS AND DISCUSSIONS

Unconsolidated porous media were reconstructed through the Monte Carlo simulation. Volume rendering of a sample particle packing and pore-space thus created are shown in figures 10(a) and (b), respectively.

Five normally distributed particle sizes were used as input parameters in the porous media reconstruction simulation. The particles used had same mean diameter of  $250\mu$  but different variances ranging from 0.01 to 100. After the particle packs are simulated, the bulk porosity and average number of contacts were calculated through image processing. The effects of particle size distribution width on these computed properties, which are relevant to porous media transport, are shown in figures 11, and 12. As the particle diameter variance increased, the porosity decreased and coordination number increased. This resulted from occupation of spaces between bigger particles filled by smaller particles. Reconstructed porous media were also generated using the same simulation method, using particle size distributions with mean diameter of  $100\mu$ ,  $250\mu$ ,  $500\mu$ ,  $750\mu$ , and  $1000\mu$ , while keeping the variance constant at 1. The bulk porosity and average number of contacts in these five packed beds were then computed using the previously used image processing methods. The variation of bulk porosity and average number of contacts with the particle size are shown in figure 13 and 14. As the particle mean size increased, the bulk porosity decreased and average contact number increased. These examples show how the approach adopted in this work can be utilized to do virtual experiments in order to predict porous media transport behavior.

The three-dimensional image of a random packing of glass beads of  $500\mu$  diameter as obtained by x-ray microtomography for validation of the packing simulation algorithm is shown in figure 15. The bulk porosity was measured as 0.35, and 0.48 for the  $250\mu$  and  $500\mu$  diameter glass bead packs, respectively. The pore size distributions of the two glass bead packs were extracted from the x-ray microtomographic 3D images using the image processing algorithms developed in this work. The extracted pore size distributions are shown in figure 17 and 19.

The size distribution of glass beads of mean diameter  $250\mu$  and  $500\mu$  were obtained through microscopic image acquisition, and automated image analysis (as described in the methods section). The size distributions thus obtained are shown in figure 5 and 6. These distribution parameters were used as input values in the Monte Carlo porous media reconstruction simulation algorithm. The pore space data matrices from these reconstructed porous media were then run through the skeletonization algorithm (described in the methods section) to extract the pore network. Three-dimensional rendering of a sample pore network is shown in figure 11. The spherical bright regions in the image depict the pores, and the bright thin structures joining them represent the skeleton of the throats. The  $250\mu$  and  $500\mu$  reconstructed porous media systems were simulated in triplicate and the pore size and tortuosity distributions of them were computed using the image processing steps described in the methods section. The calculated bulk porosities for the triplicate simulated packs were 0.35, 0.34, and 0.37 for  $250\mu$  diameter particles and 0.47, 0.51, and 0.49 for  $500\mu$  diameter particles. A t-test<sup>31</sup> comparison ( $p < 0.05$ ) found that the porosity of real glass bead pack and the simulated pack were statistically equivalent. The pore size distributions of  $250\mu$  and  $500\mu$  simulated systems are compared side-by-side with the same from the x-ray microtomographic images in figures 17 and 19. The bar-heights in case of the data from the simulation results indicate the average values and the y-error bars indicate the standard deviations from the triplicates. The differences observed overall in the pore size distributions obtained from the simulation and imaging can be attributed to the lack of sufficient contrast at the interfaces in the  $\mu$ -CT images leading to erroneous segmentation. The particle packing algorithm that was used in this work was computationally less expensive, but it did not solve the equations of motion for each particle move and interaction, unlike the ballistic drop

approach<sup>17</sup>, as all the particles were falling downward. This could introduce an overall difference in prediction of pore size distributions from what is obtained by imaging of real glass bead packs. A quantitative comparison between the pore size distributions obtained from simulated packs and the x-ray  $\mu$ -CT images were done by Kolmogorov-Smirnov statistics test. The cumulative distribution functions used in the calculation of the Kolmogorov-Smirnov statistics are shown in figure 18 and 20. The value of Kolmogorov-Smirnov statistics were 0.031 and 0.023 for 250  $\mu$  and 500 $\mu$  systems, respectively, which were less than 0.085865, the table value of  $w_{1-\alpha}$  quantile with the level of significance ( $\alpha$ ) as 0.05. Hence, the two pore size distributions are statistically equivalent.

The average tortuosity (the ratio of total length of a throat and the shortest distance between the starting and the end point of the throat) of the systems with 250 $\mu$ , and 500 $\mu$  mean diameters were obtained as 3.97 and 2.04, respectively.

#### *Limitations of the Simulation Algorithm*

Although the packing simulation results agree well with the three-dimensional imaging data there are some limitations to this method. The current algorithm can pack particles of spherical shape only. The particles and the container walls are strictly assumed to be hard, so no deformation was allowed. But in real life, many instances can be found where the constituent particles of a porous material are neither spherical nor perfectly hard. The non-spherical particle situations could be handled by considering geometric constraints related to other shapes while calculating the interaction energy in each iteration of the simulation. The soft boundary condition can be realized by defining a permissible shift window for the container boundary. But these approaches were not attempted in this work,

because the simulation was compared with real glass beads packed in a plastic cuvette both of which had hard boundaries.

## CONCLUSIONS

Porous media is encountered in numerous fields of interest ranging from water resource engineering to process industries. Hence their prediction of transport properties is crucial. Network based modeling of various porous media transport phenomena does not depend on empirically obtained parameters, and accounts for microstructural influence on macroscopic behavior. Hence proper extraction of the pore network becomes instrumental in obtaining a realistic prediction. Mapping of the network based on pore-space topology essentially needs the three-dimensional pore-space data set. To obtain this data set through an imaging instrument, the porous media must be created first. Thus only those porous structures which can be created in the laboratory can be utilized in the network based prediction model. But, by means of the current work, a wide range of porous media can be reconstructed, followed by their detailed morphological analysis extracting microstructural parameters relevant to transport modeling. Using this method, virtual experimentation can be done with a wide array of possible structures before a very few of them are short-listed for desired transport properties and validated through laboratory experiments. It will not only save time, money and other resources needed in carrying out extensive laboratory experiments in the early stages of product development, but also will provide a systematic approach for new porous media based product development.

## ACKNOWLEDGEMENTS

We acknowledge the help of Tom Dufresne of Health Care Research Center, Proctor & Gamble Co., Cincinnati, OH in obtaining three-dimensional x-ray microtomographic image of the random bead pack. We also acknowledge the help of Dr J.D. Miller of University of Utah, Salt Lake City, UT in understanding the details of the Monte Carlo simulation.

#### LITERATURE CITED

1. Bear J, Buchlin J-M. Modeling and applications of transport phenomena in porous media (1st edition). Dordrecht, The Netherlands: Kluwer Academic Publishers, 1991.
2. Wannanen KM, Litchfield JB, Okos MR. Classification of drying models for porous solids. *Drying Technology*. 1993;11:1-17.
3. Whitaker S. Simultaneous heat, mass and momentum transfer in porous media: a theory of drying. *Advances in Heat Transfer*. 1977;13:217-255.
4. Geankopolis CJ. Transport processes and unit operations (3rd edition). New Jersey: Prentice Hall P T R, 1993.
5. Gilliland ER, Sherwood TK. The drying of solids VI: diffusion equations for the period of constant drying rate. *Ind. Eng. Chem*. 1933;25:1134-1141.
6. Masmoudi W, Prat M, Bories S. Drying: percolation theory or continuum approach - some experimental evidences. In: Quiantard M, Todorovic M. *Heat and Mass Transfer in Porous Media*. New York: Elsevier, 1992:817-28.
7. Rogers JA, Kaviany M. Funicular and evaporative-front regimes in convective drying of granular beds. *Int. J. Heat Mass Transfer*. 1992;35:469-478.

8. Prat M. Recent advances in pore-scale models for drying of porous media. *Chemical Engineering Journal*. 2002;86:153-64.
9. Neimark AV. Multiscale percolation systems. *Sov. Phys. JETP*.1989;69:786–91.
10. Broadbent SR, Hammersley JM. The mathematics of percolation in networks. *Proc. Cambridge Philosophical Society*. 1957;53:629-35
11. Prat M, Bouleux F. Drying of capillary porous media with stabilized front in two-dimensions. *Phys. Rev. E*. 1999;60:5647-56.
12. Fredrich JT. 3D imaging of porous media using laser scanning confocal microscopy with application to microscale transport processes. *Phys. and Chem. of the Earth*. 1999;24:551-61.
13. Lindquist WB, Venkatarangan A. Investigating 3D geometry of porous media from high resolution images. *Phys. and Chem. of the Earth*. 1999;25:593-99.
14. Tastevin G, Nacher PJ. NMR measurements of hyperpolarized  $^3\text{He}$  gas diffusion in high porosity silica aerogels. *Journal of Chem. Physics*. 2005;123:064506-1:13
15. Oh W, Lindquist WB. Image thresholding by indicator Kriging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1999;21:590-601.
16. Sherwood JD. Packing of spheroids in three-dimensional space by random sequential addition. *J. Phys. A: Math. Gen*. 1997;30:L839-L43.
17. Coelho D, Thovert J-F, Adler PM. Geometrical and transport properties of random packing of spheres and aspherical particles. *Phys. Rev. E*. 1997;55:1959-78.
18. Mort P, Riman RE. Determination of homogeneity scale in ordered and partially-ordered mixtures. *Powder Technology*. 1995;82:94-104.

19. Williams SR, Philipse AP. Random packings of spheres and spherocylinders simulated by mechanical contraction. *Phys. Rev. E.* 2003;67:051301-1-9.
20. Cheng YF, Guo SJ, Lai HY. Dynamic simulation of random packing of spherical particles. *Powder Technology.* 2000;107:123-30.
21. Lam L, Lee S-W. Thinning methodologies – a comprehensive survey. *IEEE transaction on pattern analysis and machine intelligence.* 1992;14:869-85.
22. Siddiqi K, Bouix S. Hamilton-Jacobi skeleton. *Int. J. of Computer Vision.* 2002;48:215-31.
23. Bouix S, Siddiqi K, Tannenbaum. Flux driven automatic centerline extraction. *Medical Image Analysis.* 2005;9:209-21.
24. Ma CM, Sonka M. A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding.* 1996;64:420-33.
25. Lee T-C, Kashyap RL, Chu C-N. Building skeleton models via 3-D medial surface/axis thinning algorithms. 1994;56:462-78.
26. Wu QJ, Bourland JD. Three-dimensional skeletonization for computer-assisted treatment planning in radiosurgery. 2000;24:243-51.
27. Hafford KJ, Preston KJ. Three-dimensional skeletonization of elongated solids. *Computer Vision, Graphics, and Image Processing.* 1984;27:78-91.
28. Maddah M, Soltanian-Zadeh H, Afzali-Kusha A, Shahrokni A, Zhang ZG. Three-dimensional analysis of complex branching vessels in confocal microscopy images. *Computerized Medical Imaging and Graphics.* 2005;29:487-98.
29. Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *J. Opt. Soc. Am. A.* 1984;1:612-9.

30. Conover WJ. Practical Nonparametric Statistics(1st edition). New York: Wiley, 1998.
31. NIST/SEMATECH e-handbook of statistical methods,  
<http://www.itl.nist.gov/div898/handbook/>

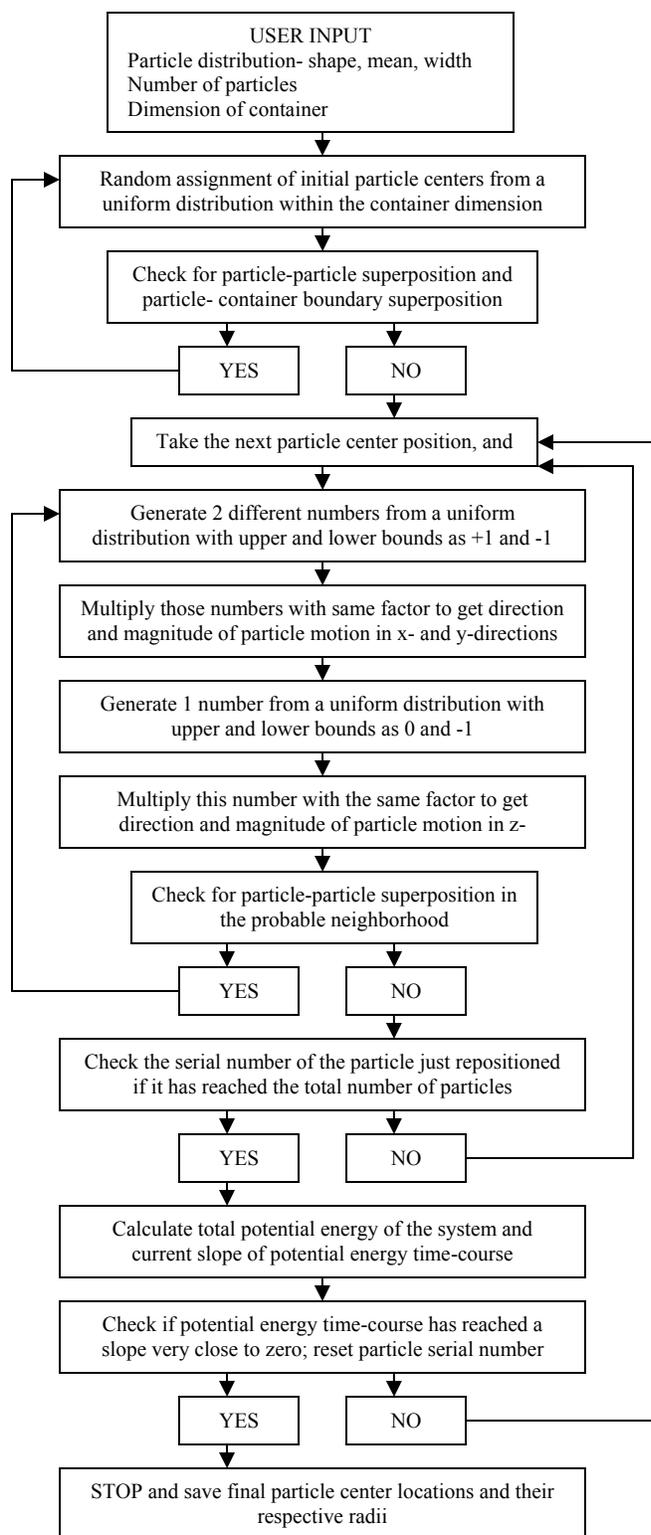


Figure 1: Monte Carlo gravitational particle packing simulation algorithm

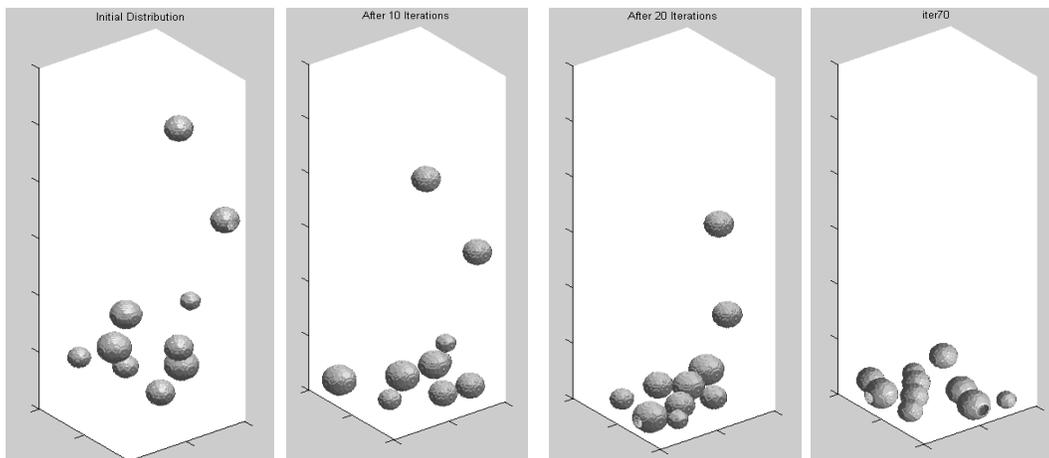


Figure 2: Example of movement of particles undergoing Monte Carlo gravitational simulation

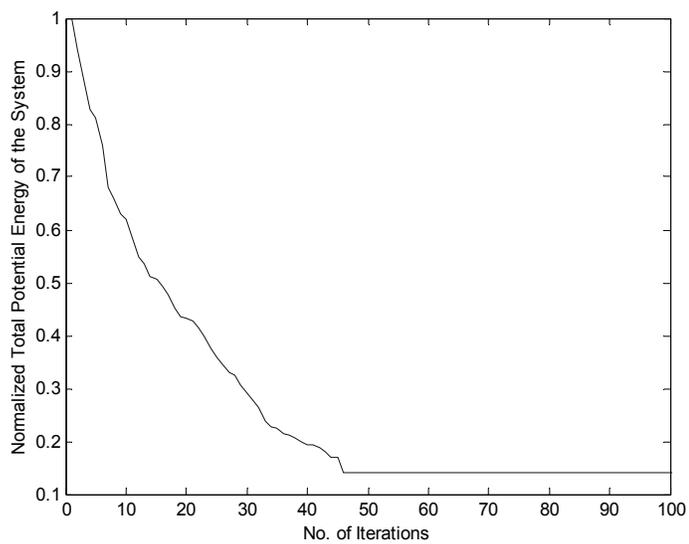


Figure 3: Normalized total potential energy of the particle system as the simulation progresses

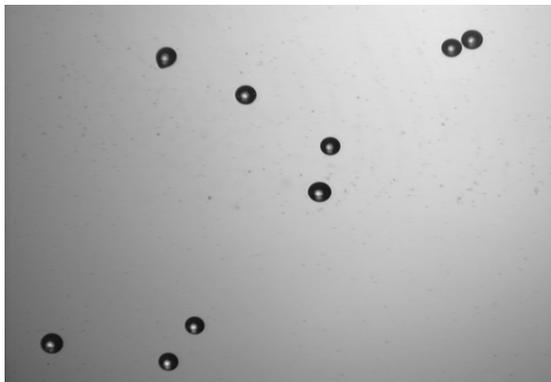


Figure 4: An example of acquired microscopic images of glass beads

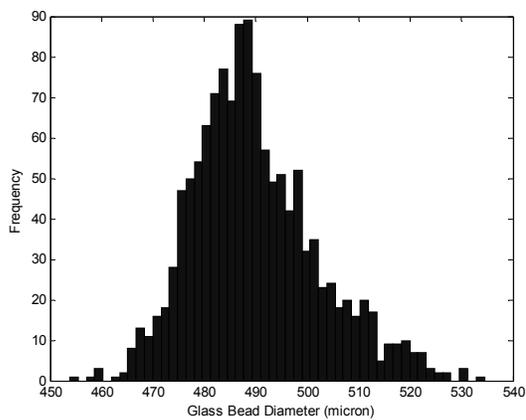


Figure 5: Size distribution of 500μ diameter glass beads used in the experiment

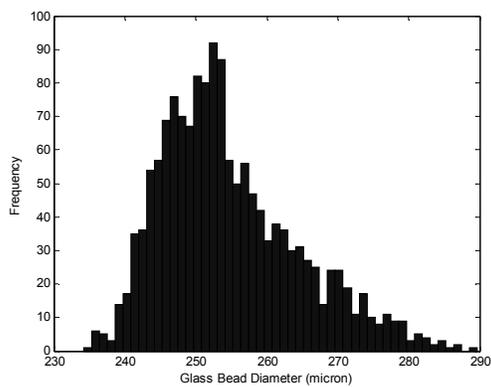


Figure 6: Size distribution of 250μ diameter glass beads used in the experiment

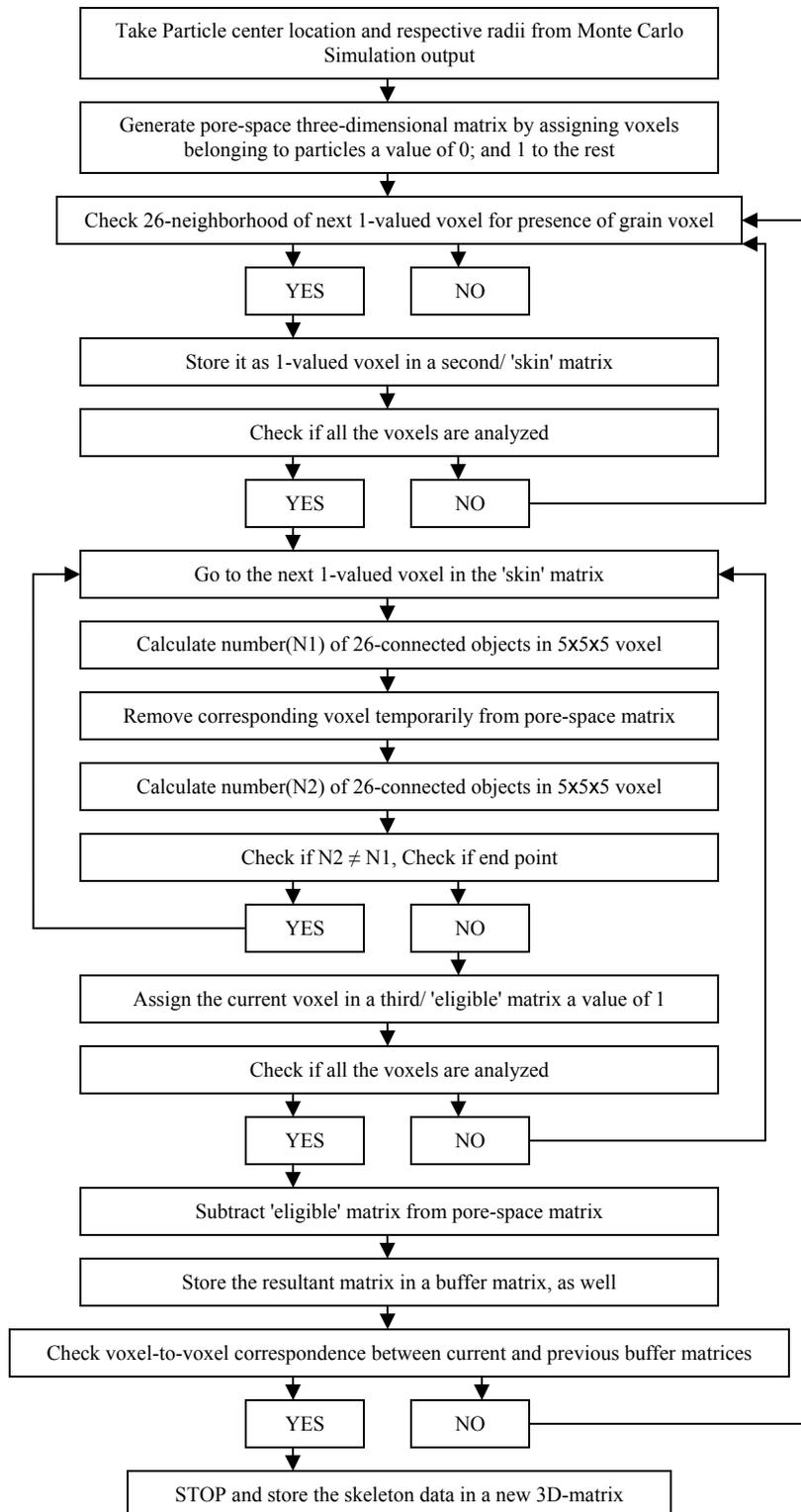


Figure 7: Flow chart of three-dimensional skeletonization algorithm to extract pore network from simulated unconsolidated porous media

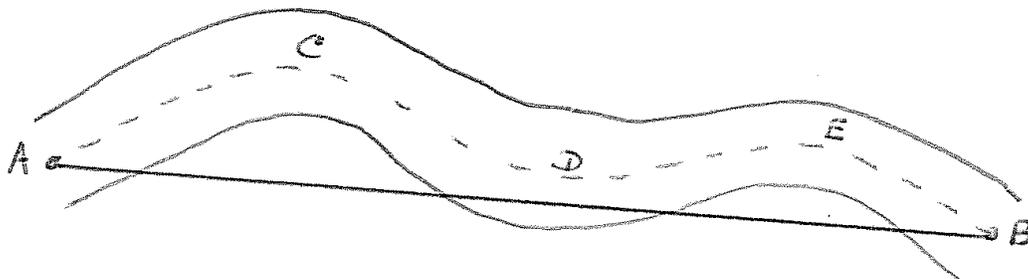


Figure 8: Explanation of the definition of tortuosity used in this manuscript. The tortuosity is the ratio of the path-length through the points ACDEFB and the straight line distance AB.

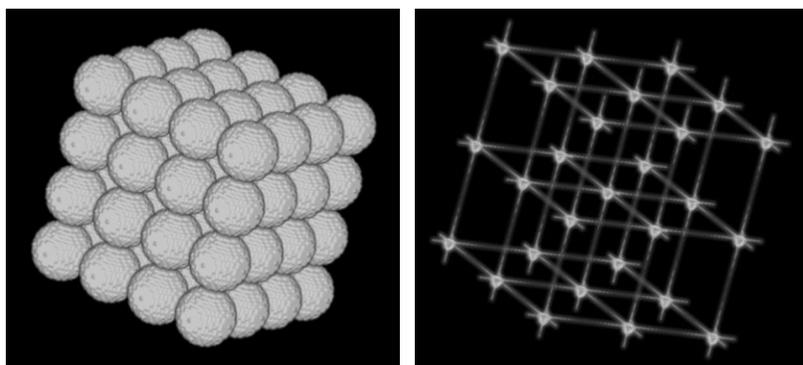


Figure 9: (a) Cubic lattice arrangement of simulated bead pack(on the left); (b) three-dimensional skeleton extracted using current algorithm(on the right)

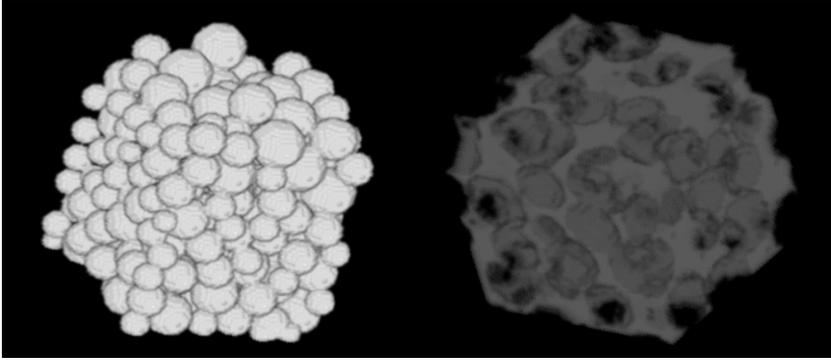


Figure 10: (a) Volume rendering of a Monte Carlo simulated reconstructed porous medium (on the left); (b) Volume rendering of the pore-space (on the right)

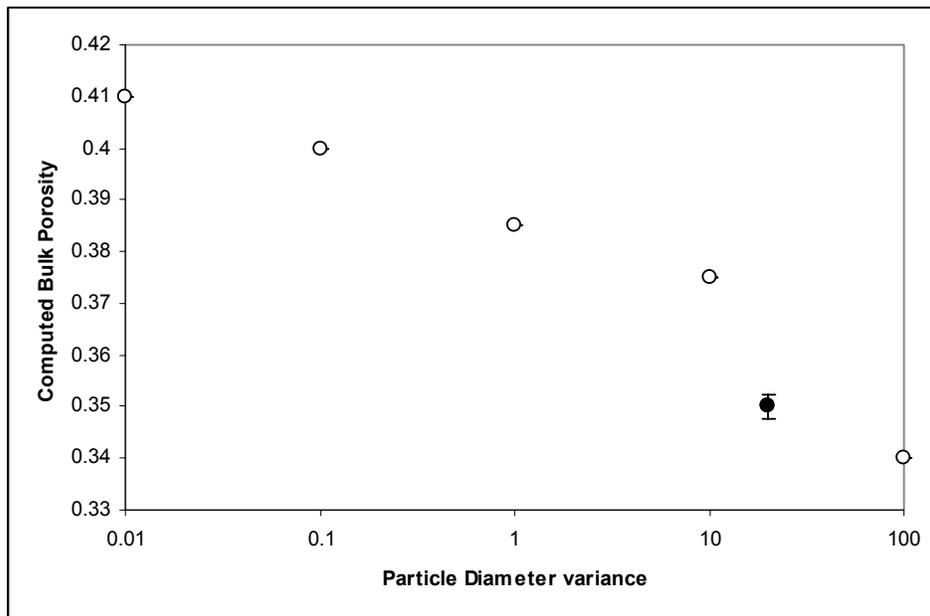


Figure 11: Influence of particle size distribution width on bulk porosity obtained through simulation. The filled circle corresponds to the real pack of 250 $\mu$  diameter beads, y-error bar indicates standard deviations from triplicate simulations.

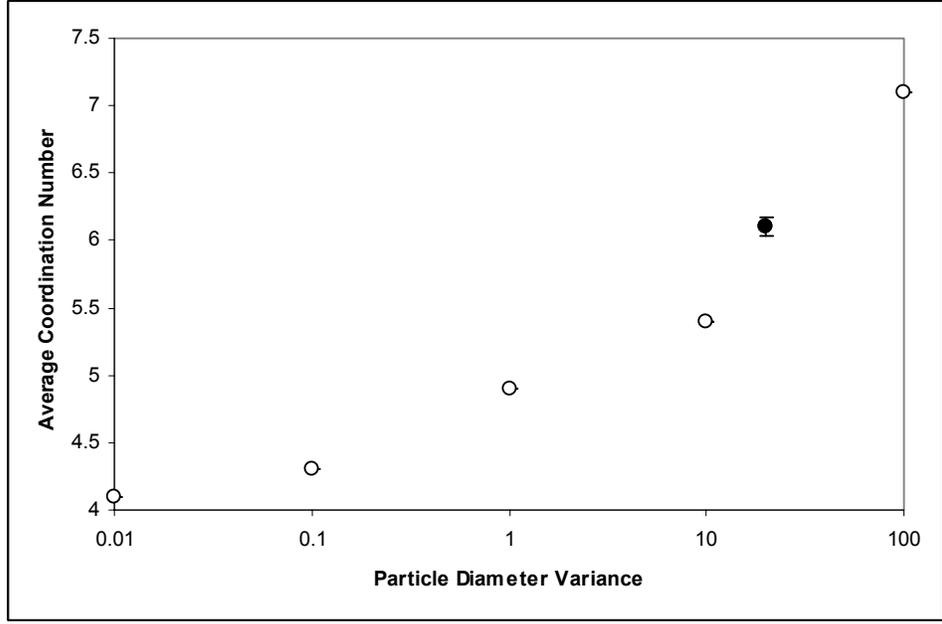


Figure 12: Influence of particle size distribution width on coordination number obtained through simulation. The filled circle corresponds to the real pack of 250 $\mu$  diameter beads, y-error bar indicates standard deviations from triplicate simulations.

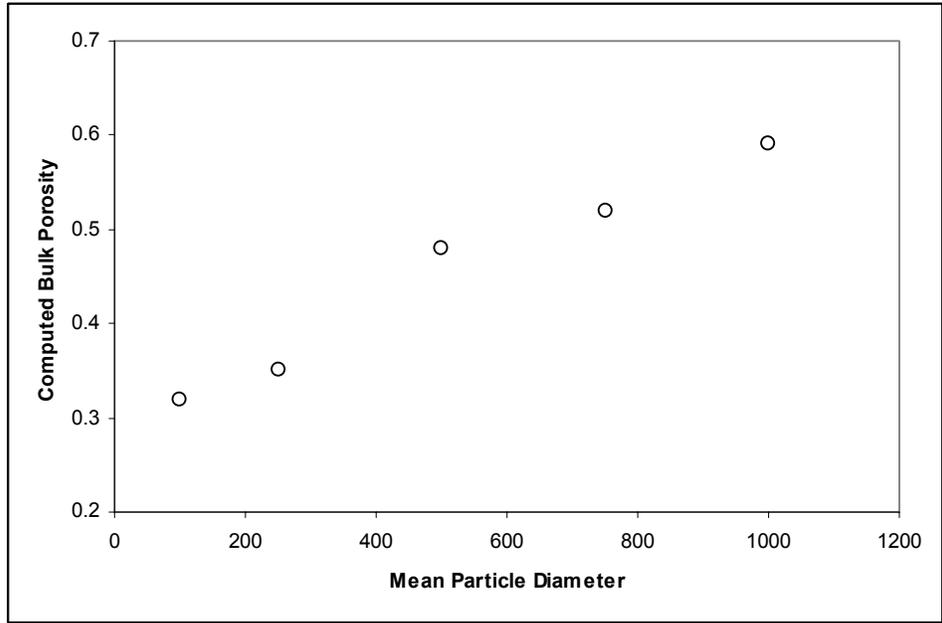


Figure 13: Influence of mean particle size (with constant variance of 1.0) on bulk porosity obtained through simulation.

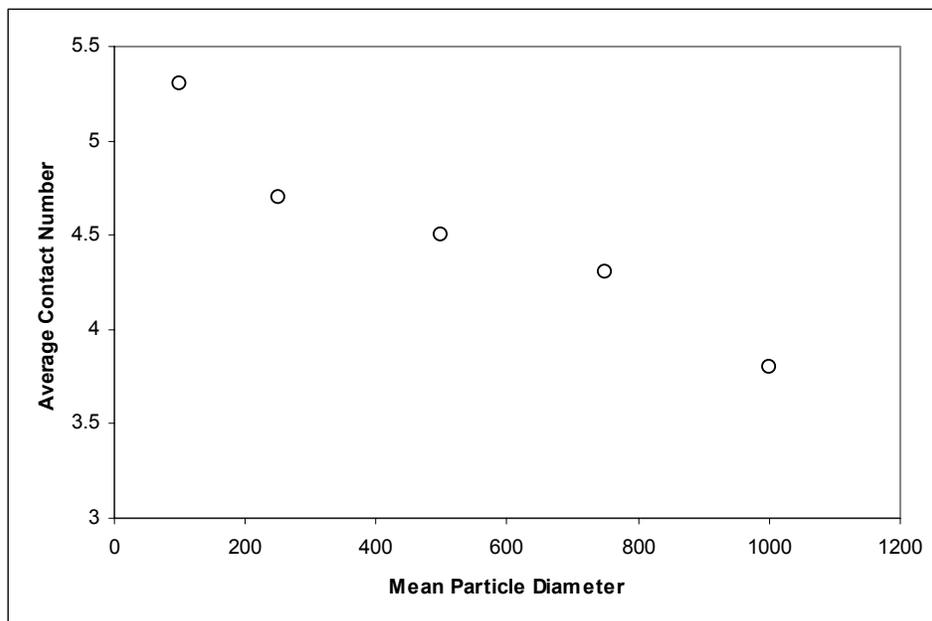


Figure 14: Influence of mean particle size (with constant variance of 1.0) on coordination number obtained through simulation.

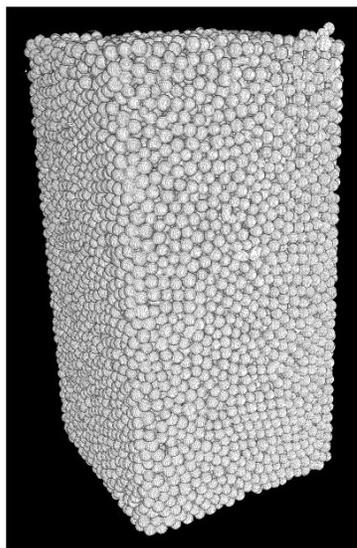


Figure 15: Volume rendering of x-ray microtomographic image of random packing of glass beads

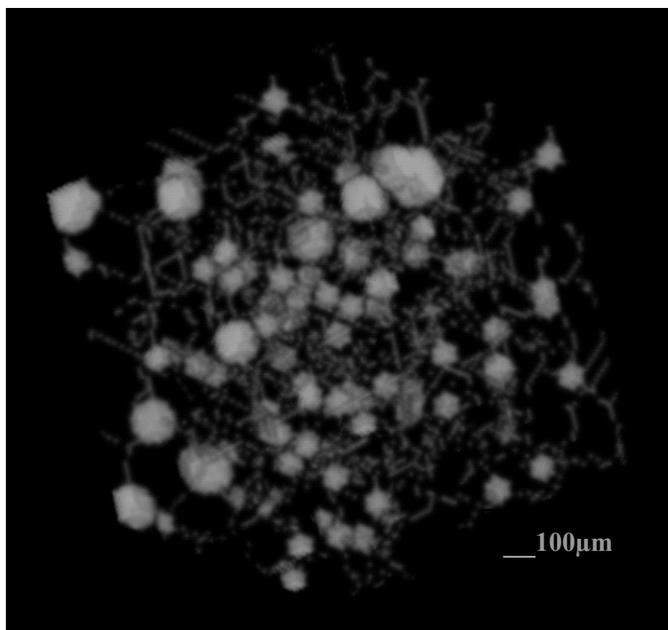


Figure 16: Three-dimensional rendering of a sample extracted pore-network

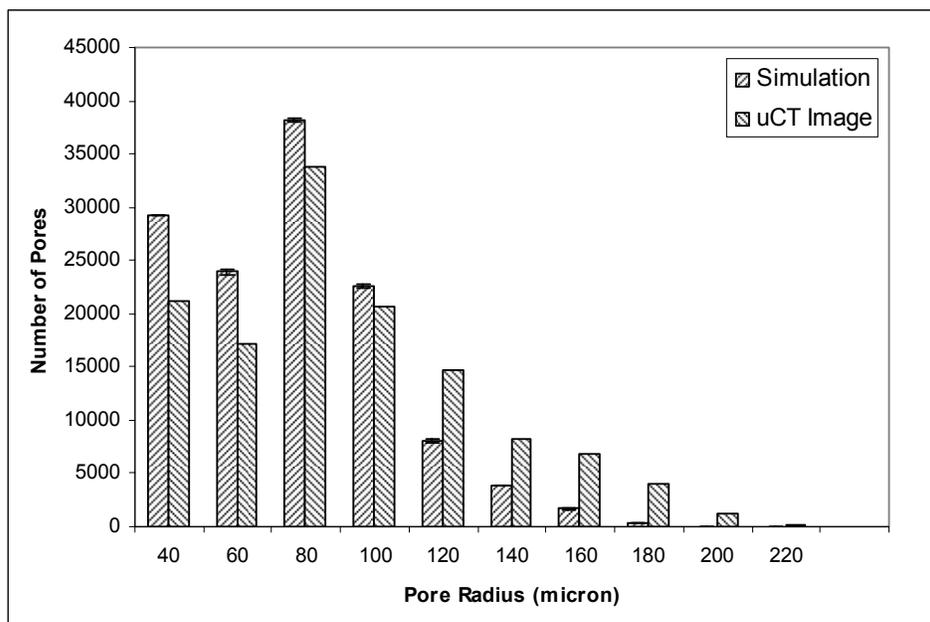


Figure 17: Pore size distributions obtained from simulated packed beds and  $\mu$ -CT image of 250 $\mu$  beads. The y-error bars indicate the standard deviations in triplicate simulations.

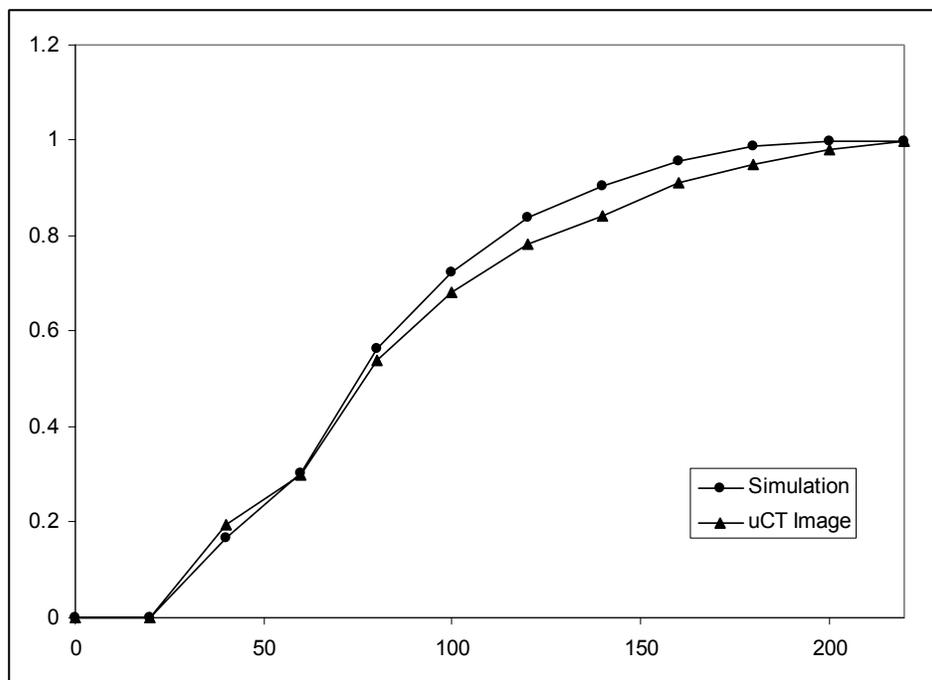


Figure 18: Cumulative distribution functions of pore size distribution of 250 $\mu$  bead pack; from packing simulations and  $\mu$ -CT image. This is used in the Kolmogorov-Smirnov statistics calculation.

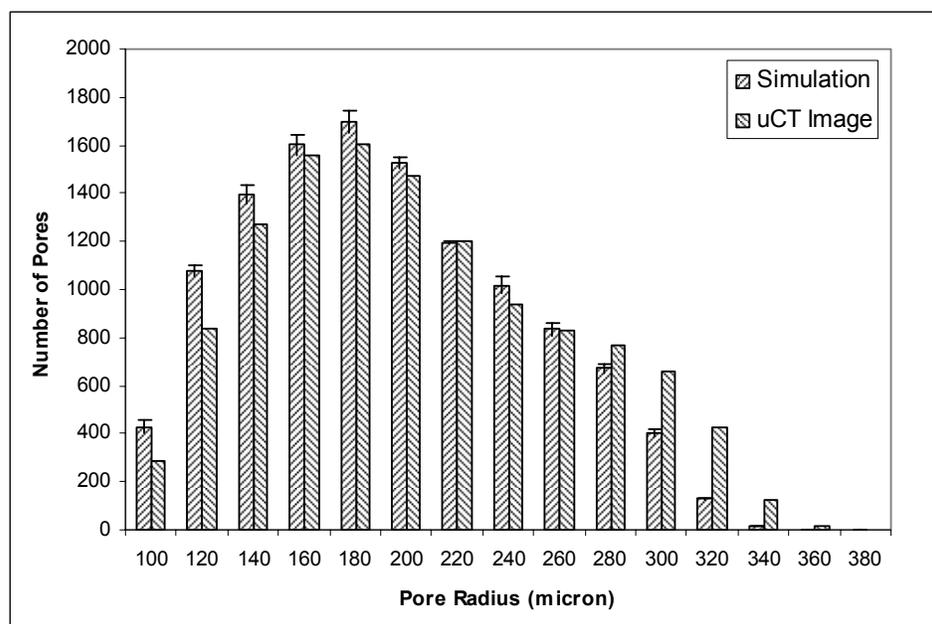


Figure 19: Pore size distributions obtained from simulated packed beds and  $\mu$ -CT image of 500 $\mu$  beads. The y-error bars indicate the standard deviations in triplicate simulations.

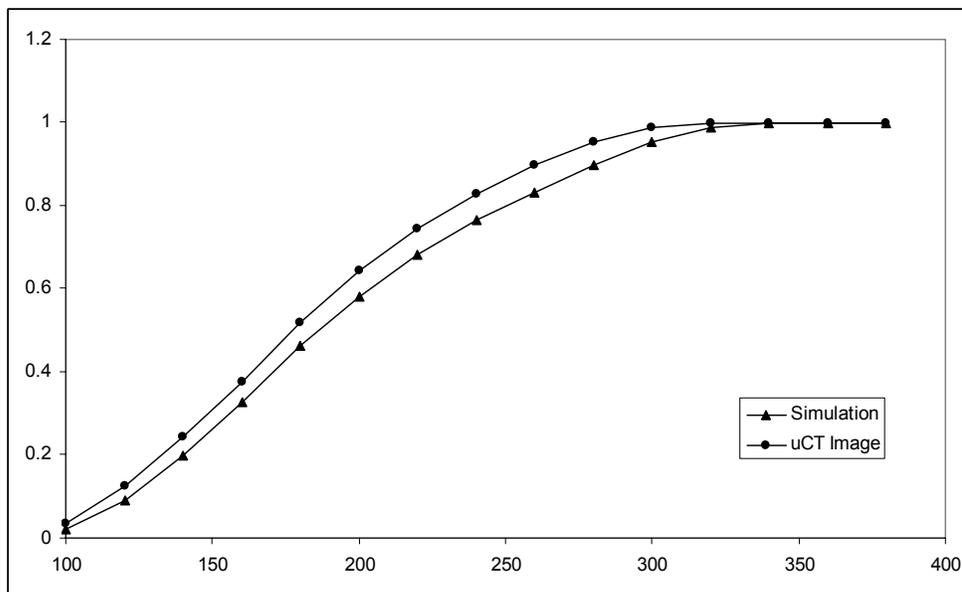


Figure 20: Cumulative distribution functions of pore size distribution of 500 $\mu$  bead pack; from packing simulation and  $\mu$ -CT image. This is used in the Kolmogorov-Smirnov statistics calculation.

## CHAPTER IV

### TITLE:

A Simulation Based Method to Assess Inversion Algorithms for Transverse Relaxation Data

### AUTHORS:

Supriyo Ghosh, Kevin M. Keener, Yong Pan

KEY WORDS: Relaxation, inverse problem, simulation

### ABSTRACT:

NMR relaxometry is a very useful tool for understanding various chemical and physical phenomena in complex multiphase systems. A Carr-Purcell-Meiboom-Gill (CPMG) [1] experiment is an easy and quick way to obtain transverse relaxation constant ( $T_2$ ) in low field. Most of the samples usually have a distribution of  $T_2$  values. Extraction of this distribution of  $T_2$ s from the noisy decay data is essentially an ill-posed inverse problem. Various inversion approaches have been used to solve this problem, to date. A major issue in using an inversion algorithm is determining how accurate the computed distribution is. A systematic analysis of an inversion algorithm, UPEN [2, 3] was performed by means of simulated CPMG data generation. Through our simulation technique and statistical analyses, the effects of various experimental parameters on the computed distribution were evaluated. We converged to the true distribution by matching up the inversion results from a series of true decay data and a noisy simulated data. In addition to simulation studies, the same approach was also applied on real experimental data to support the simulation results.

### INTRODUCTION:

NMR relaxometry is a very useful tool for understanding various chemical and physical phenomena in complex multiphase systems. One of the key parameters that are highly exploited to indirectly measure several properties of the sample is the transverse relaxation time ( $T_2$ ). This time constant is measured most commonly by a simple CPMG experiment [1]. In a CPMG experiment, echo heights are usually recorded at discrete linearly spaced time points and the decay of the echo heights are characterized by the time constant  $T_2$ , where time dependent echo signal can be represented as,

$$S(t) = S_0 \exp(-t/T_2) \quad (1)$$

Here,  $S_0$  is echo envelop height at initial time point,  $S(t)$  is echo height as a function of time, and  $t$  is the time vector. When the sample under investigation has only one type of species in identical environments the echo heights would obey a mono-exponential decay pattern. But in most systems, there are several different species in multiple environments. All these ‘species-environment’ combinations, in principle, will have characteristic  $T_2$  values for their decaying echo signals. It is intuitive that, in natural systems, the transition from one  $T_2$  value to the next would rather be smooth. This results in a  $T_2$  distribution with multi-components (bi-modal, tri-modal, etc.). Thus, the discrete-time decay data obtained through the CPMG experiment on a complex sample will have information regarding all the components of the entire distribution hidden in it. This raw data will be corrupt by some noise, as well. Hence, to have a clear understanding of a sample through NMR relaxometry, it is crucial to obtain a reliable deconvolution of all the  $T_2$  components from the discrete, noisy, decaying echo signal. This is essentially an ill-conditioned inverse problem, which can be represented in the form of well known Fredholm integral equation of first kind (IFK) [4],

$$S_d = S_0 \int_a^b \exp(-t/T_2) f(T_2) dT_2 + \varepsilon \quad (2)$$

where,  $S_d$  is signal raw data,  $S_0$  is maximum signal,  $f(T_2)$  is the frequency of a species with a specific characteristic  $T_2$ , and  $\varepsilon$  is random noise.

Since the CPMG data collected in an experiment is discrete in time the IFK here is better represented in the discrete form of,

$$S_i = S_0 \times \{\sum f_k(T_2) \exp(-t_i/T_{2k})\} + \varepsilon \quad (3)$$

It is characteristic of this type of IFK equations that it is difficult to obtain useful solutions to them by straightforward methods like Gaussian elimination, Cholesky, or QR-Factorization.

This inverse problem of  $T_2$  deconvolution has been solved by linearization, followed by the use of different kinds of regularization parameters [5, 6]. Several physically convincing constraints like non-negativity are also introduced to dampen the unnecessary oscillations in the solutions. The Uniform-Penalty inversion (UPEN) algorithm is one of the most successful regularization processes applied to NMR relaxation data [2, 3]. This algorithm is fundamentally a least square minimization routine. To avoid excessive details in the computed distribution a penalty factor is also added to the squared error of fit, in this algorithm. A curvature penalty factor is adopted in UPEN within the data range. A coefficient ( $C_k$ ) is also multiplied to the penalty factor. The value of this coefficient,  $C_k$  is iteratively adjusted in order to ensure a strict uniformity of the penalty factor. Thus, the main aim of this algorithm is to minimize,

$$\sum (S_0 \times \sum f_k(T_2) \exp(-t_i/T_{2k}) - S_i)^2 + \sum C_k (f_{k-1}(T_2) - 2f_k(T_2) + f_{k+1}(T_2))^2 \quad (4)$$

where the first term is the residual term and the second term is the penalty term. Similar to any other inverse problem cases, this deconvolution process has the drawback of non-uniqueness. Even if there exists an exact solution (i.e. a specific distribution of  $T_2$ ) that has been used to generate the raw decay data, it may not be unique. The same raw data can

satisfy several different model solutions very nicely (with very low magnitude of error). Rank-deficiency in the data is usually a major contributor to this problem [4]. Thus, the estimated model may be significantly smoothed or otherwise biased relative to the true model. Characterizing such bias through model resolution analysis is essential to properly applying models to real systems. The ill-conditioned nature of the inverse problem imparts instability into the computed solution. A small change in a single measurement can lead to an enormous change in the estimated model. Thus, as the noise level rises the computed model loses its precision in resolution. Because of these reasons, it is essential to perform systematic statistical analyses on the responses of a regularization algorithm like UPEN to a range of CPMG data similar to real experiments. If we synthesize data from a known model of  $T_2$  distribution and then try to calculate back the true model using a regularization algorithm, it will provide an estimate of accuracy of the algorithm. This study defines limits on the UPEN algorithm based on a known  $T_2$  distribution and a specific set of experimental conditions.

#### METHODS:

An algorithm is developed and the code is written in Matlab<sup>®</sup> (can be obtained by requesting the first author) to generate simulated CPMG  $T_2$  exponential data. This algorithm takes the user inputs in terms of: number of components, relative contribution of each component, shape of each component, mean  $T_2$  values of each component, variance of each component, and signal-to-noise (SNR) of the measured data. All raw discrete-time data are normalized. SNR effect is studied by creating simulated CPMG data with SNR level from 80 through 1000. The computed model resolution is studied by creating simulated CPMG data (with commonly encountered SNR level of 400) with component mean separations ranging

from 450 through 900ms. UPEN algorithm is run on these simulated data to obtain the computed distributions. The deviation of the computed solution from the true model is quantitatively analyzed through two-sample Kolmogorov-Smirnov test. The Kolmogorov-Smirnov test statistic is the maximum of absolute differences between two cumulative distribution functions of the distributions being compared [7]. If  $F^*(x)$  is the true distribution and  $F(x)$  is the computed distribution the hypotheses for our two-sided test would be:

$$H_0 : F(x) = F^*(x) \text{ for all } x \text{ from } -\infty \text{ to } +\infty \quad (5)$$

$$H_1 : F(x) \neq F^*(x) \text{ for at least one value of } x \quad (6)$$

The test statistic,  $T_1$  is expressed as,

$$T_1 = \sup_x |F^*(x) - F(x)| \quad (7)$$

The decision rule is that we reject  $H_0$  at the level of significance  $\alpha$  if the test statistic,  $T_1$  exceeds  $1-\alpha$  quantile  $w_{1-\alpha}$  as given by the corresponding lookup table.

The means and standard deviation of the computed distributions were calculated using the log-normal probability distribution function:

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2} \quad (8)$$

where  $\mu$ , and  $\sigma$  are the mean and standard deviation of the variable. The computed distribution means and standard deviations were compared with gradual increase of SNR and the distance between the component means.

For experimental validation of the proposed new method, and to show the usefulness, we generated some real data, as well. Three solutions of  $\text{MnSO}_4$  of varying strengths were made so that their  $T_2$  values are 1 order of magnitude apart. Three different NMR tubes of diameter of 5mm were filled with these  $\text{MnSO}_4$  solutions, and placed together in a 15mm

diameter tube for generation of CPMG data using the minispec mq series bench-top NMR equipment tuned at 26MHz (Bruker Optics Inc., TX, USA). We did the same experiment with the same sample with increasing number of scans, to observe the effect of SNR on computed distribution. We measured the SNR in the CPMG data with 128 scans. We generated simulated CPMG data from a model distribution with three components whose mean values correspond to that of the three different  $\text{MnSO}_4$  solutions obtained from the data with very high SNR (2048 scans). We added noise to it of the same level as measured from the 128 scan experiment (SNR = 256). Similarly, we generated more simulated data with gradual increase in the component width. We did UPEN inversion on all these simulated data. We calculated the Kolmogorov-Smirnov statistic between the distribution computed by UPEN from the real data (128 scans), and the same from the simulated noisy data with varying component widths.

## RESULTS:

Simulated data were created for one component  $T_2$  distribution with mean value of 100ms and standard deviation of 5ms. SNR values of 40, 60, 80, 100, 200, 300, 400, 500, and 1000 were used to generate the simulated CPMG data. As the SNR reduced, the broadening of the computed peak is observed. Two extreme cases are shown by comparing with the true model in Figure 1. The two-sample Kolmogorov-Smirnov tests are done between the true distribution and computed distributions obtained from simulated CPMG data with increasing SNR levels. The Kolmogorov-Smirnov statistic values are plotted against SNR levels in Figure 2, to show how SNR contributes towards the deviation of computed model from the true model. For UPEN, we observed that if we achieve SNR level of about 400 that gave us

best estimation of the  $T_2$  distribution; we do not need to increase the SNR beyond that value. The relationship between SNR and the departure of the standard deviation (width of the peak) values of computed distribution from that of the true distribution was calculated and is shown in Figure 3. It is observed to show a negative power law type relationship.

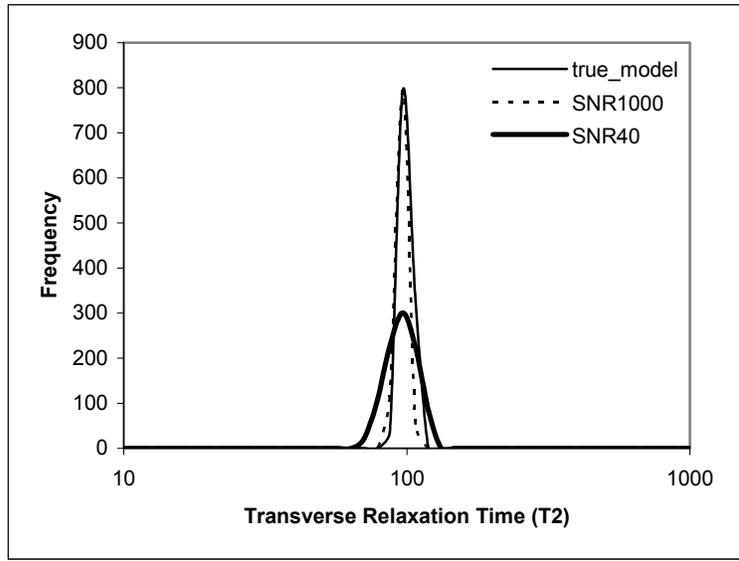


Figure 1: Comparison of true  $T_2$  distribution and UPEN computed  $T_2$  distributions from data with varying noise levels

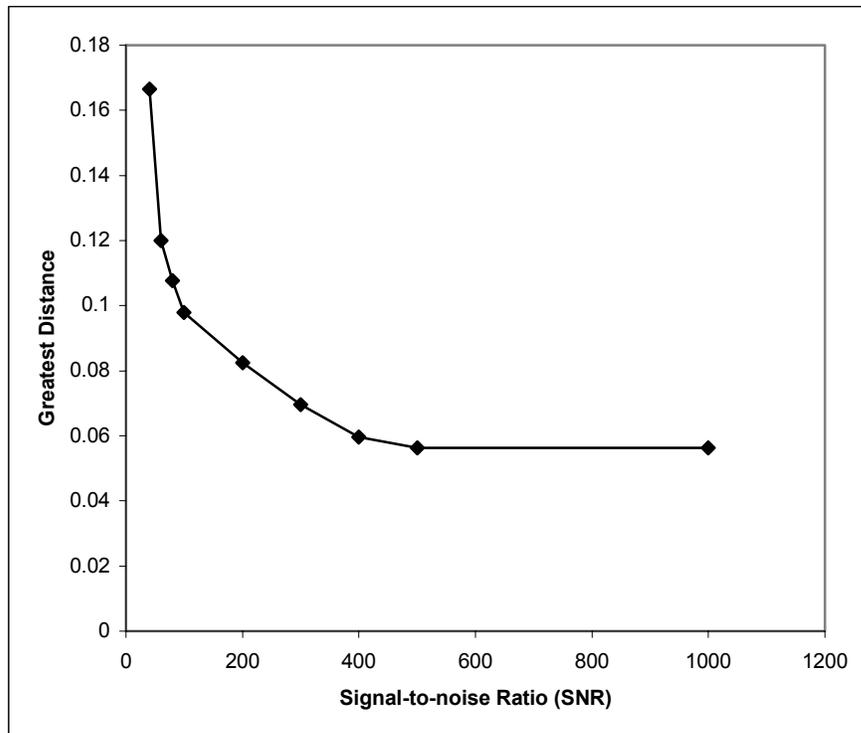


Figure 2: Effect of signal-to-noise in raw data on deviation of computed mean  $T_2$  value from the true mean  $T_2$  value

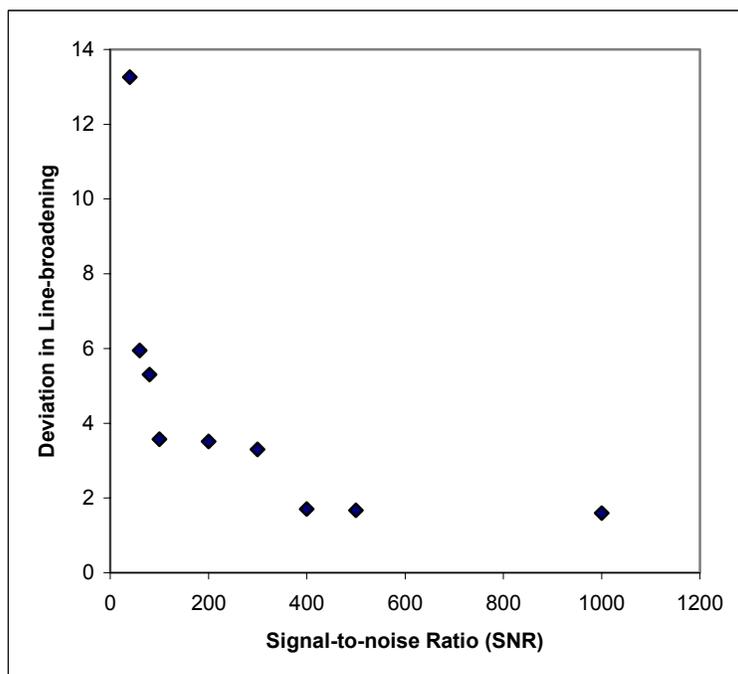


Figure 3: Deviation in computed  $T_2$  distribution width due to signal-to-noise ratio

Simulated data were created for two component  $T_2$  distributions. The distance between the component-means are set at 500, 550, 600, 700, and 900ms. The standard deviations of each component were 5. The SNR level is kept constant at 400 for all simulations. This level of SNR is chosen because the Kolmogorov-Smirnov statistic value is observed to almost flatten out beyond 400 SNR. The evolution of the computed distribution as the two components approach each other is shown in Fig. 4. Kolmogorov-Smirnov test is done between the true and computed distributions.

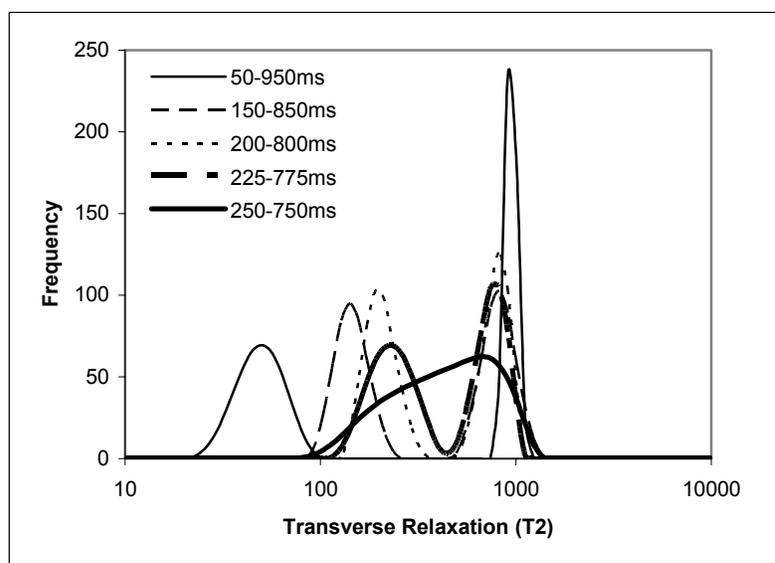


Figure 4: Resolving efficiency of UPEN algorithm (while signal-to-noise is kept constant)

The UPEN computed distributions from real experimental data of  $MnSO_4$  solutions are shown in Fig.5. The increased resolution of components as the number of scans i.e. SNR increases is evident from the figure.

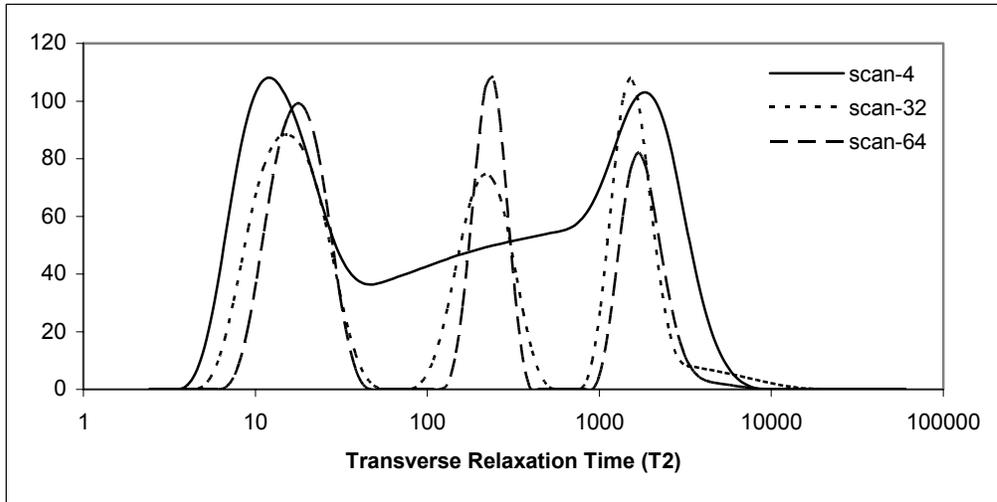


Figure 5: Effect of signal-to-noise ratio on the peak resolving efficiency

The calculated values of Kolmogorov-Smirnov statistic between the distribution computed by UPEN from the real experimental data (128 scans), and the same from the simulated noisy data with varying component widths are plotted in Fig.6.

The width of the model distribution was increased from 0 to 2 with linear intervals of 0.25. The width of 1.25 is observed to give the lowest value of Kolmogorov-Smirnov statistic. The value of Kolmogorov-Smirnov statistic in this case is 0.052 which is less than 0.06072 the table value of  $w_{1-\alpha}$  quantile with the level of significance ( $\alpha$ ) as 0.01. as a result, we can accept the null hypothesis i.e. the computed distribution can be considered equivalent to the true distribution. Hence, we can conclude that the true distribution of  $T_2$  in the sample will have a width very close to this value.

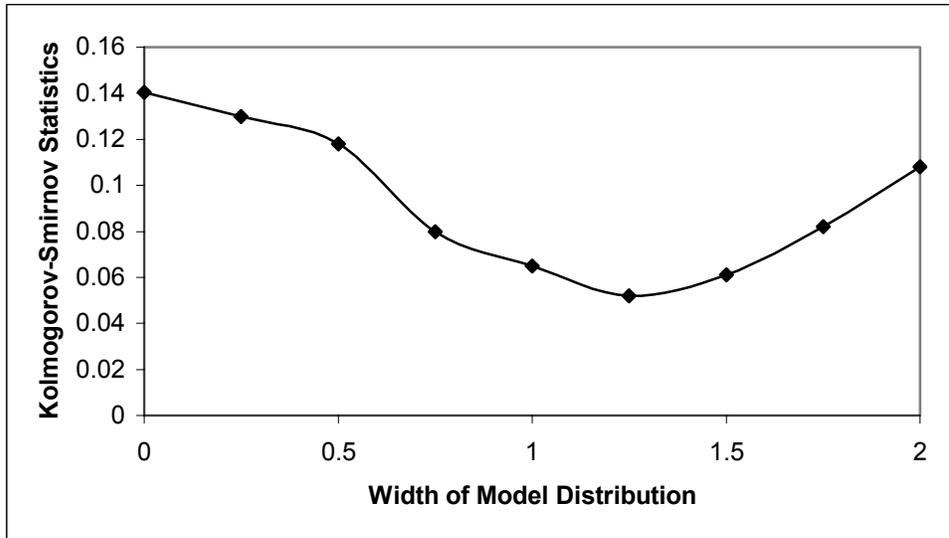


Figure 6: Evolution of Kolmogorov-Smirnov statistics between true distribution and computed  $T_2$  distribution as starting distribution width is gradually changed keeping the signal-to-noise ration constant

#### DISCUSSION:

A major issue in using inversion algorithms for deconvolution of  $T_2$  data, is that when we get a computed distribution we cannot determine how accurate the solution is or in other words how close the computed distribution is to the true distribution. Variation in several parameters like SNR, number of  $T_2$ s we are solving for, data spacing, etc. have influence on the final solution. For many real systems we have no ‘*a priori*’ knowledge of the  $T_2$  distribution. Thus we cannot determine solution accuracy.

Following our approach of using simulated CPMG data to evaluate the efficiency of the deconvolution achieved by an inversion algorithm, we were able to do detailed quantitative statistical comparison between the true distribution and the computed distribution, and quantify the limitations of the algorithm. The discrete-time decay data is corrupt by noise before the inversion algorithm is operated. Just like any other inverse problems, small variation in raw data introduced appreciable oscillation in the computed

model, resulting in broadening of the peaks. From Fig. 3 we can see a power law type trend in the departure from the true standard deviation as SNR changes. Using the Kolmogorov-Smirnov test, we could observe that as the SNR approaches 400 level and increases beyond that, the difference between the computed and true model flattens out and falls below the quantile table value for significance level of 0.10. From this observation we can conclude that achievement of an SNR level of 400 can give us enough accuracy in the computed distributions generated by UPEN algorithm.

The bi-modal simulation studies showed us that the UPEN algorithm can distinguish two different components (with 5 standard deviations) until their mean values are separated by at least  $\sim 550$ ms. This conclusion is valid only for an SNR level of 400, which we kept fixed during this part of simulation study. 400 SNR level being commonplace in many NMR relaxometry studies it can be concluded that if two populations of  $T_2$ s are  $\geq 550$ ms apart then the computed distributions obtained through UPEN algorithm are reliable.

By matching up the real data with varying width simulated data, we could also converge to a very good estimate of the true  $T_2$  distribution in an unknown sample. From this analysis we conclude that, when we have data with a specific SNR level, the model distribution which had the lowest Kolmogorov-Smirnov statistic value would be the closest prediction for the true distribution of  $T_2$  in the real sample. This way we were able to alleviate the uncertainty in the estimated distribution computed by an inversion algorithm, even in the absence of complete '*a priori*' knowledge of the sample.

This approach of systematic evaluation of regularization algorithms used for deconvolution of  $T_2$  data, enabled us to quantify the accuracy of the computed distributions, and helped us reduce the uncertainty in the estimated distribution when very little is known

about the sample. This study also provides guidance regarding the extent of SNR we need to achieve to get a reliable distribution. The approach described here can also be very useful in optimizing the regularization parameter for a specific kind of relaxation data.

#### ACKNOWLEDGEMENTS:

We acknowledge the help of Dr. R.J.S. Brown and Dionne Swift for sharing the UPEN algorithm TrueBASIC code. We also acknowledge the help we received through the discussions regarding statistical analysis with Dr Jason A. Osborne.

#### REFERENCES:

- [1] P.T. Callaghan, Principles of Nuclear Magnetic Resonance Microscopy, Clarendon Press, Oxford, 1991.
- [2] G.C. Borgia, R.J.S. Brown, P. Fantazzini, Uniform-penalty inversion of multiexponential decay data, *Journal of Magnetic Resonance*, 132 (1998) 65-77.
- [3] G.C. Borgia, R.J.S. Brown, P. Fantazzini, Uniform-penalty inversion of multiexponential decay data II. data spacing,  $T_2$  data, systematic data errors, and diagnostics, *Journal of Magnetic Resonance*, 147 (2000) 273-285.
- [4] R.C. Aster, B. Borchers, C. H. Thurber, Parameter Estimation and Inverse Problems, Elsevier Academic Press, New York, 2005.
- [5] W. Provencher, A constrained regularization method for inverting data represented by linear algebraic or integral equations, *Computer Physics Communications* 27 (1982) 213-227.
- [6] G.C. Borgia, V. Bortolotti, R.J.S. Brown, P. Castaldi, P. Fantazzini, U. Sovneri, A comparison among different inversion methods for multi-exponential NMR relaxation data, *Magnetic Resonance Imaging*, 12 (1994) 209-212.

[7] W.J. Conover, Practical Nonparametric Statistics, Wiley, New York, 1998.

## APPENDICES

## TITLE:

Study of porous media drying using simulated porous media, pore network drying, and magnetic resonance imaging

## AUTHORS:

Supriyo Ghosh, Kevin M. Keener, Michael Bills, Yong Pan

## ABSTRACT:

Drying in porous media is encountered in numerous situations ranging from water resources management to industrial processes of varying length scales. Porous media drying has been described using continuum mechanics, for many decades. But this approach fails to predict the drying behavior properly because it does not account for the microstructural influence. This method also needs to use experimentally obtained transport coefficients, which makes it semi-empirical. In this study, porous media are reconstructed through three-dimensional Monte Carlo simulation, followed by mathematical morphology based image processing to extract the pore network. This extracted pore network provided the microstructural information, on which an invasion percolation based drying simulation was applied. The simulation results were compared with in-situ magnetic resonance imaging of drying packed bed; with respect to fractal dimensions of saturated regions, drying curves, and evolution of wet patches. Statistical comparison between the time-series data of fractal dimensions in case of experimental data and simulation showed equivalence at the p-value of 0.05. The drying curves from simulation and gravimetric experiment agreed well for the first part of drying. The evolution pattern of wet patches also showed similar pattern in experiment and simulation.

## INTRODUCTION:

Drying is a process in which moisture in a substance is vaporized and subsequently removed from the surface. It is frequently encountered as a unit operation in industrial processes (e.g. food processing, chemical technology, textile, etc.). The drying behavior of a material also dictates its processing as well as functional attributes. It helps determine what the optimum parameters for drying operation should be set at in order to make the process most economical. The texture and other mechanical properties of the dried material are also influenced by the drying pattern. Most materials that undergo drying are porous in structure. Hence, the ability to describe the process of drying as accurately as possible is crucial in having more control over the process and achieving the desired product characteristics after the drying step.

Modeling of the drying of porous media has been an active field of research for decades. However, even the simplest situation, such as the drying of a bed of glass beads at room temperature, is still a challenging problem for the modelers. Traditionally, the problem is treated within the framework of continuum approach to porous media.<sup>1</sup> This approach relies on the generalized Darcy's law and the relative permeability concept, and the heterogeneous multiphase system of porous media is considered as a fictitious continuum. As discussed by Masmoudi et al.<sup>2</sup>, the relevance of this traditional approach has not yet been fully demonstrated. The shortcomings of the continuum approach like its inability to lead to a general method of predicting the interfacial mass transfer coefficients for drying, inspired the investigation of drying with a microscopic perspective: the percolation modeling.<sup>3</sup> By modification of the invasion percolation algorithm for drying, they were able to simulate drying in pore-scale in two-dimensions and support it satisfactorily with simplistic experimental data.<sup>4,5</sup> But the percolation process differs dramatically from two-dimensional

structure to three-dimensional one;<sup>6</sup> and the real-life porous media that undergoes drying in the industry are all made of three-dimensional pore structure. An attempt has been made to extend this simulation to three-dimension.<sup>7</sup> However, these simulations are based on artificially fixed parameter distributions to characterize the pore structure. Also, in these simulations, pores have been assumed to be at the corners of a perfect cubic lattice, and they are connected by straight throats. Hence these simulation results cannot satisfactorily describe and predict porous media drying in real life. Building a realistic porous bed using simulations and then extracting the pore space information for use in the invasion percolation algorithm renders the method more close to real-life porous media drying, and thus gives us more confidence in predictions. A non-invasive, quantitative as well as high resolution method to probe the phase evolution during drying of porous media can provide a strong evidence to support the simulation results. Powerful image processing pipelines utilizing mathematical morphology based three-dimensional skeletonization and template based dilation, can extract the pore network parameters from the three-dimensional pore-space data.<sup>8</sup> Hence, the relevant approach in understanding the drying of porous media in greater detail is to reconstruct porous media with control over the grain size distribution using a simulation close to real-life situation; extract parameters of the simulated pore-space using image processing sequence; simulate drying using the extracted parameters with the help of invasion percolation method.

Mainly two types of pore-scale modeling are found in literature: Lattice-Boltzmann (LB) and pore network models. LB models, utilizing concepts originally introduced with lattice-gas models, are capable of simulating flow and transport in the actual pore space.<sup>9,10,11</sup> Increasing computational power has allowed for more realistic LB simulations. However, these models

are very complex and there is limited data for quantitative comparisons. Pore network models are effective tools used to investigate or predict macroscopic properties from fundamental pore-scale behavior of processes and phenomena based on geometric volume averaging. Pore network models are mechanistic models that utilize an idealization of the complex pore space geometry of the porous media. This is usually achieved by representing the pore space by pore elements having simple geometric shapes. For example, pore-bodies and pore-throats have been represented as spheres and cylinders or cubes and cylinders, respectively. Network models have been used in the fields of chemical engineering, petroleum engineering and hydrology to study a wide range of single and multiphase flow processes. They have been used to study relative permeability<sup>12,13,14</sup>; the effect of pore structure on relative permeability and capillary pressure hysteresis in two phase systems<sup>15</sup> prediction of permeability and hydraulic conductivity<sup>16</sup>; investigation of the functional relationship between capillary pressure, saturation, and interfacial areas<sup>17</sup>; prediction of permeability and residence time distributions for mechanical dispersion in packed beds<sup>18</sup>; drainage and imbibition<sup>19</sup>; phase distributions, interfacial areas, and mass transfer<sup>20</sup>; ganglion formation and mobilization<sup>21,22,23</sup>. However, there are only very limited studies using network modeling to investigate phase change phenomena like drying in porous media. Satik and Yortos<sup>24</sup> studied bubble growth and nucleation as a result of pressure reduction or heat transfer. Lattice gas cellular automata has also been utilized to simulate evaporation in two-dimensional porous media. McCall and Guyer<sup>25</sup> simulated equilibrium evaporation phenomena on a square bond lattice by lowering the vapor pressure. Prat and co-workers<sup>26,27,28</sup> modeled drying as an invasion percolation process in which the viscous force is assumed to be negligible compared to the capillary force. They performed simulations on two- and three-dimensional set-up, and

also performed experiments in ideal two-dimensional set-up. They tried to qualitatively as well as quantitatively show the phase distribution and drying curves through this process. Agreement between the experimental and simulation-based studies was very good in two-dimensions but not as satisfactory while trying to explain the experimental data in real three-dimensional porous media. Several reasons can be put forth explaining the discrepancies those were observed. Two-dimensional simulation of network modeling can never be close to the experimental results involving real porous media. This is because, the connectivity in two-dimension is very different from the one in three-dimension, dramatically influencing the critical percolation values or percolation threshold. Thus using an experimental method to non-invasively look-into the three-dimensional phase distribution in the porous media during the time course of drying, extracting the quantitative data from those three-dimensional data can really assist develop as well as evaluate the invasion percolation model of drying of porous media. In invasion percolation the invasion of sites at each time step of the simulation is decided by the invasion potential of the sites located at the interface between the invading and the defending fluids. A physical basis can be given to the magnitude of this invasion potential by relating it to the capillary pressure. The capillary pressure,  $\Delta P_{\text{cap}}$ , across a liquid-liquid interface in a tube is:

$$\Delta P_{\text{cap}} = 2(\sigma \cos \theta) / R \quad (1)$$

where  $\sigma$  is the interfacial tension between the two fluids,  $\theta$  is the contact angle, and  $R$  is the radius of the tube. There are two different immiscible displacement processes: imbibition and drainage. During drainage, a non-wetting fluid displaces a wetting fluid, which is similar to the drying process. Since, the dimension of the drying bed used in this study is very small (1cm x 1cm x 2cm), the gravitational effect can be neglected. In the absence of gravity,

immiscible displacement, like drying of saturated porous bed, can be characterized by the dimensionless capillary number,  $C_a = (\text{viscosity of displaced fluid} \times \text{average velocity of displaced fluid}) / \text{interfacial tension}$ .  $C_a$  basically represents the ratio of viscous forces to capillary forces. In the case of drying, for air as the displacing fluid and water as the displaced one, typically  $C_a$  is smaller than  $1 \times 10^{-10}$ .<sup>2</sup> It indicates that, the drying system studied in this work is dominated by the capillary effect. According to capillary pressure equation above, the capillary forces are strongest at the narrowest places in the porous medium. Therefore, if all the pore throats are smaller than the pore bodies, the invasion of the pore body will lead to spontaneous invasion of the adjacent pore throats. Hence, the radii of the pore throats are the controlling factor in drying. The capillary pressure is less in bigger pore bodies, and hence those will be invaded first. In this case the capillary pressure established in the pore bodies and pore throats will resist the invasion of the non-wetting fluid. The capillary forces are stronger at throats than in pore bodies. The non-wetting fluid invades the pore bodies more easily than the pore throats and thus drainage is controlled by the pore throats, with the larger pore throats being more easily invaded. Hence, in case of drainage, the invasion potential is inversely proportional to the capillary pressure in the pore throats. Broadbent and Hammersley<sup>29</sup> first proposed the theory of percolation in the year of 1957. Percolation essentially describes the accessibility of the network or lattice. The sites in a lattice or the bonds between these sites can be either open or closed with respect to transport through them. IPM was first proposed by Chandler et al.<sup>30</sup> and first implemented by Wilkinson and Willemsen<sup>31</sup> on various regular two-dimensional and three-dimensional lattices both with and without trapping of the defender fluid. Each site were invaded one at a time after each time increment in the simulation, the site invaded being the one with the

highest invasion potential of all sites at the liquid-liquid interface. Therefore, invasion percolation is essentially a dynamic form of the percolation theory. As noted by these authors, although IPM is motivated by the liquid-liquid displacement process, it is applicable to any invasion process following the path of least resistance. Wilkinson and Willemsen<sup>31</sup> demonstrated that when trapping of the defending fluid was included a significant difference was revealed between the two-dimensional and three-dimensional lattices, the clusters of trapped defender fluid being considerably larger for the two-dimensional lattice. This result shows a limitation of the two-dimensional lattice to model realistic porous systems that are usually inherently three-dimensional. Wilkinson<sup>32</sup> and Glass and Yarrington<sup>33</sup> later extended the IPM simulation of Wilkinson and Willemsen to include the effect of gravity as the stabilizing and destabilizing force. Shaw<sup>34</sup> first suggested that drying can be considered as an invasion percolation process. As drying is generally a slow process, capillary forces dominate over viscous forces. The air invades the pore throat with the lowest capillary pressure, i.e. the pore throat with the widest width. The pore body, normally wider than the pore throat, is automatically invaded once the adjacent pore throat is invaded. This forms the basis of the invasion percolation model of drying. Pat and his co-workers<sup>26,27,28</sup> recognized the similarities between drainage and drying, and started to model drying as an invasion percolation process. In both drainage and drying, the liquid within the medium is the defender fluid while the air is the invader fluid. However, the air and the liquid exit the medium in the same direction in drainage and in opposite directions in drying. Another major difference drying and conventional invasion percolation is that in drainage, isolated clusters of liquid can become trapped and cannot be invaded further due to fluid incompressibility; clusters of liquid become isolated from the main bulk but gradually disappear under the

action of evaporation. Prat and his co-workers considered these dissimilarities. They also adopted some reasonable simplifying assumptions: the liquid is a single component liquid, the gas phase is a binary mixture of vapor of the liquid and inert air, Kelvin effect can be neglected, diffusive transport is quasi-steady, vapor escapes through the top edge of the network, zero flux conditions prevail across the other three boundaries of the porous medium.

The current work extends this approach one step further in simulating drying in porous media that would be more realistic. Instead of assuming a specific pore size distribution and location of the pore centers as cubic lattice points, a more realistic pore network directly extracted from the three-dimensional pore-space data set is used as the input in the invasion percolation algorithm. The tortuosity (the ratio of total length of a throat and the shortest distance between the starting and the end point of the throat) factor of the throat connecting the pores is also introduced. The simulation data is supported by non-invasive phase distribution mapping by in-situ magnetic resonance imaging. The pore-networks which were utilized in the drying simulation were obtained through image processing of the pore-space reconstructed using Monte Carlo gravitational simulation of particle packing. This approach opens up the possibility of performing virtual experiments on porous media drying by choosing a wide array of particle size distribution, which constitutes the packing. This will reduce the amount of resources needed to carry out numerous laboratory experiments in porous media based product design. Extensive evaluation can be done of the end-product characteristics before only very few most promising structures are short-listed for experimental assessment.

## METHODS:

### *Surface Characterization:*

The glass beads those were used in this experiment were acid washed and silanized in order to remove any hydrophilic or hydrophobic ligands from the surface. The contact angle of deionized water with the glass bead surface was one of the input parameters of the drying simulation. Hence the contact angle was measured experimentally. Two glass capillaries were taken. One of them was given the same surface treatment as the glass beads. Deionized water was drawn into them through capillary action. The water column menisci inside those capillary tubes were then examined using a microscope and digital images were acquired (Fig. 1). The contact angles were then calculated manually on the images and compared. As the images in Fig.1 depict, the contact angle for the surface treated glass was very close to  $90^\circ$ .

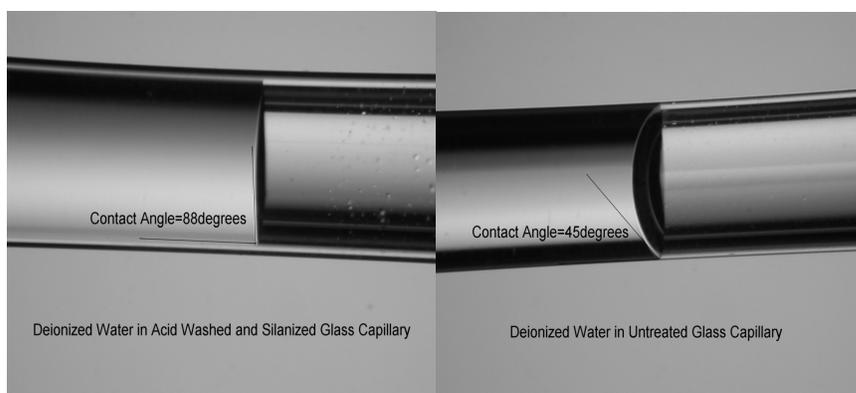


Figure 1: Microscopic images of contact angles before and after surface treatment

### *NMR Imaging Experiment:*

In order to support the drying simulation result, a drying cell was made that fits inside the magnetic resonance imaging (MRI) coil so that image acquisition can be done while drying is taking place without altering any experimental parameters. The drying cell was made of Teflon which does not produce any nuclear magnetic resonance (NMR) signal at the

radio frequency coil of the instrument. The drying cell was so constructed that the air stream passes along the outside walls of the cuvette, which is filled with saturated glass bead pack, and then passes over the open top surface of the packed bed. The drying bed had a dimension of 10mmx10mmx20mm. This construction ensured isothermal drying condition. Figure 2 provides the details of the construction. The glass beads used in this work are obtained from Sigma Aldrich ( St Louis, MO), and they had mean diameters of 250 $\mu$  and 500 $\mu$ . These glass beads had their surface acid washed and silanized. Identical treatment was given to glass capillaries and the surface contact angle was measured by pulling deionized water into it followed by light microscopy. The microscopic images of glass capillary with and without the surface treatment are shown in Figure 1. The measured contact angle in the surface treated glass capillary was 88°. Hence, in this work the pore and throat surfaces are assumed as neither hydrophilic nor hydrophobic. A controlled air drying system was built in house. It had an inlet to receive air from the air compressor through and pressure control valve. The pressure was always maintained constant at 15psi. The air coming in first passed through a pair of Drierite drying columns (Laboratory Gas Drying Unit, with 1/8" SS NPT fittings, manufactured by W.A. Hammond Drierite Co. Ltd., Xenia, OH) arranged in parallel. Only one was used at any point of time while the other was being regenerated. The dry air at the outlet of these columns then passed through a flow meter and next through an air heater. The flow of air stream was kept constant through out the experiments at 0.1573 x 10<sup>-3</sup> cubic meter per second. The heated air

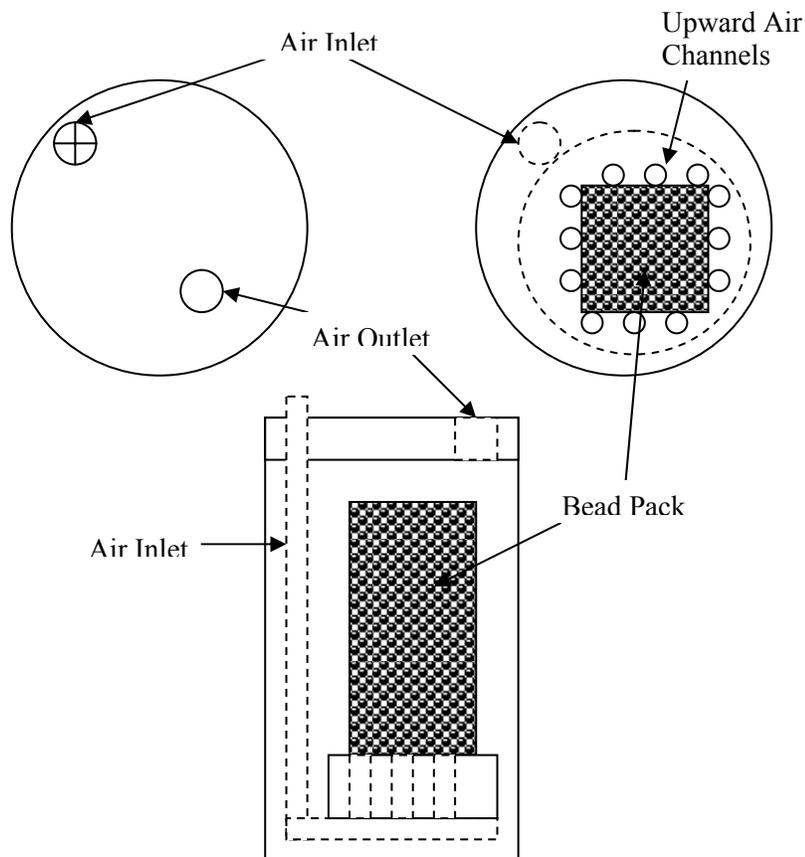


Figure 2: Drying cell for in-situ drying inside MRI instrument

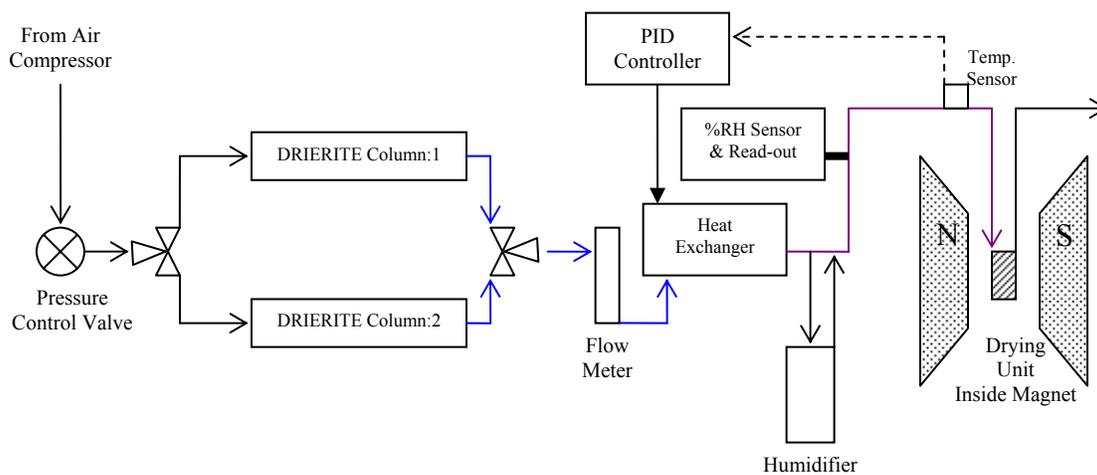


Figure 3: Controlled air drying set-up for in-situ drying of glass bead pack

line had an online temperature sensor. The temperature sensor sent feedback signal to a PID controller, which turned the air heater on and off to keep the outlet air temperature constant.

The PID controller was set to maintain the air temperature at 25°C. The heated air humidity was also checked periodically and was always found to be very close to 0%RH under stabilized condition. This drying setup is also used to dry the bead pack outside the MRI coil and gravimetric measurements were done to calculate the convective mass transfer coefficient of the system considering 0%RH boundary condition. As the stability of the air flow was achieved the air was flown through the drying cell inside the MRI instrument, Bruker Ultrashield 500MHz micro-imaging system (Bruker Biospin, Billerica, MA). The packed bed system is dried for 30minutes before each MRI image acquisition is done. Each image was acquired in two-dimension in 64x128 matrix using multi-spin-multi-echo(MSME) sequence with 12 echoes recorded per image acquisition. Since each image acquisition took about 10 minutes to run, the drying air supply was bypassed through a humidifier to achieve 100%RH before it is fed to the drying cell. This ensured no drying during image acquisition, so that we could assume a quasi-static stage during the imaging process. Images were acquired upto 8 hours of drying time. The same experiment was done for both 250 $\mu$  and 500 $\mu$  bead packs in triplicates. A program is written in Matlab® (The Mathworks Inc., Natick, MA) to automatically calculate the proton density i.e. water saturation map of the drying bead packs at all time points from the multi-echo data sets. The saturation map gray-scale images were binary gated using Otsu algorithm.<sup>35</sup>

*Gravimetric Measurement:*

Glass bead packs of both 500 and 250 $\mu$  diameters were made and dried inside the drying cell, as described in the previous section. The bead pack is measured for weight of remaining water at 0.5 hour intervals. This drying curve is stored for comparison with simulation results.

*Packing Simulation and Network Extraction:*

The glass beads which were used for the drying experiments described above were spread over a light microscope with bottom light configuration, and two-dimensional images are acquired. Images were obtained for approximately 1000 beads. An algorithm was developed to automate the process of finding the beads in those images, calculating their radii, and storing them in a data array. The algorithm is based on the principle of gating the bead images from the background, and calculating the radius from the cross-sectional area. The array was then used to determine the shape, mean, and width of the distribution of the bead sizes.<sup>8</sup> This algorithm was validated by manually measuring 200 beads under the same microscope. Now, this distribution was used as one of the input parameters while running a three-dimensional Monte Carlo simulation of particle packing, in triplicate. During this simulation the particles are assigned random initial locations in the space. In each iteration, all the particles are moved sidewise and downward without overlapping each other and without crossing the boundary. The simulation stops on its own when the change in total potential energy of the system is  $1 \times 10^{-5}$ . The packs thus obtained were converted into three-dimensional pore-space images. These binary pore-space images were run through the three-dimensional skeletonization algorithm developed in-house. The details of the simulation and image processing algorithms and their validations are given in our previous publication.<sup>8</sup> The final output of this image processing pipeline were the pore-throat network of the simulated bead packs, which had information regarding the location of pore centers, their respective radii, the throat connecting the pores, the respective minimal throat radii.

*Drying Simulation:*

The detailed pore network obtained in the previous section was the input for the drying simulation. The drying simulation was based on invasion percolation mechanism. The simulation initiates by emptying the first layer of pores in the network. In the next step onwards, for each currently empty pore the saturated pores connected to it were ascertained. The respective minimal radii, and the tortuosity of the throats which connect these saturated pores to the current empty pore were stored in two arrays. The minimal radii and tortuosity arrays were multiplied element by element. The largest element of this product array was found and the corresponding throat was determined. Now in the current invasion step, the saturated pore which was connected to the current empty pore through this throat was emptied. As one pore was emptied the corresponding pore and throat volume was dumped into a buffer variable. As the value of this buffer variable reached the convective mass transfer rate, determined by aforesaid experiment, the current time step stopped. This process continued until all the pores are emptied and the porous bed is completely dry. The steps of the algorithm are elaborated in Fig. 4.

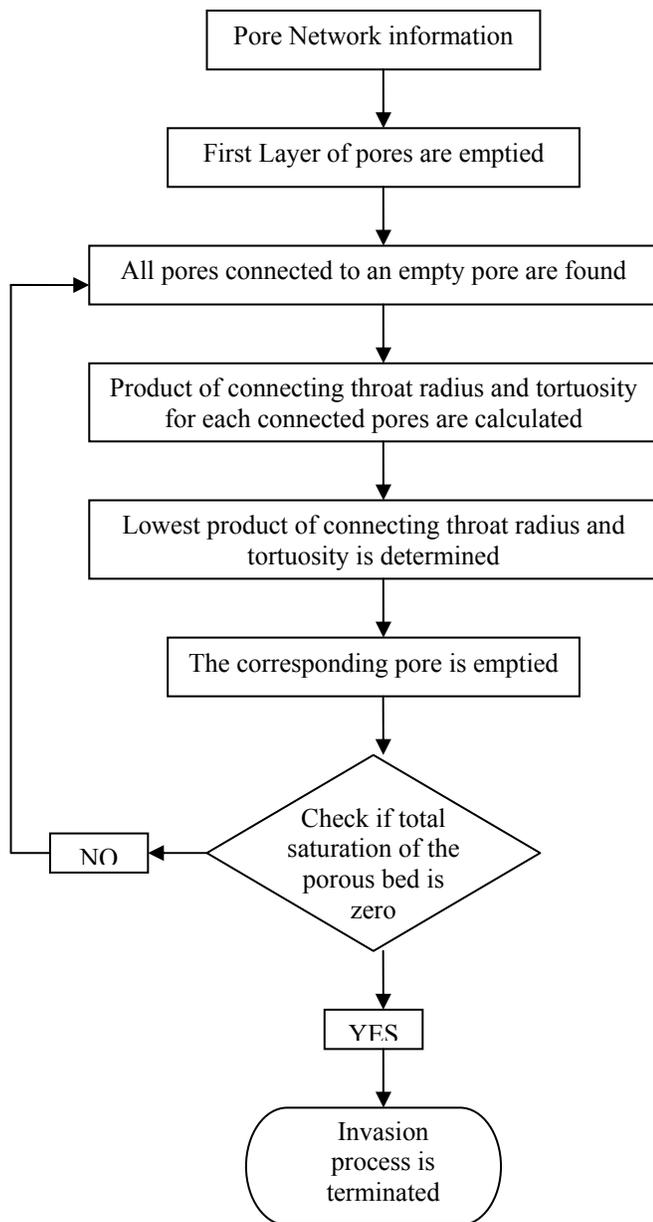


Figure 4: Invasion percolation algorithm for drying simulation

*Quantitative Image Processing:*

An image processing algorithm was developed to calculate the number of wet patches in those images. The wet patches were defined as 8-connected isolated objects which had zero-valued pixels throughout its boundary. The fractal dimensions of the saturation maps (at

all the time points during drying) were then calculated using box-counting method.<sup>36</sup> The saturation maps were first binary gated, followed by its contour extraction. The image was then divided into square boxes of pixel dimensions ranging from 2 to 32. In each case the total number of boxes that contained a part of the contour is stored. A plot is obtained for each saturation map where the x-axis is the  $\log(1/\text{block size})$  and the y-axis is the  $\log(\text{number of boxes containing the contour})$ . The slope of this plot was the fractal dimension<sup>37</sup> of the saturation distribution. The algorithm flow chart is given in Figure 5. Figure 6 shows the steps in calculating the fractal dimension.

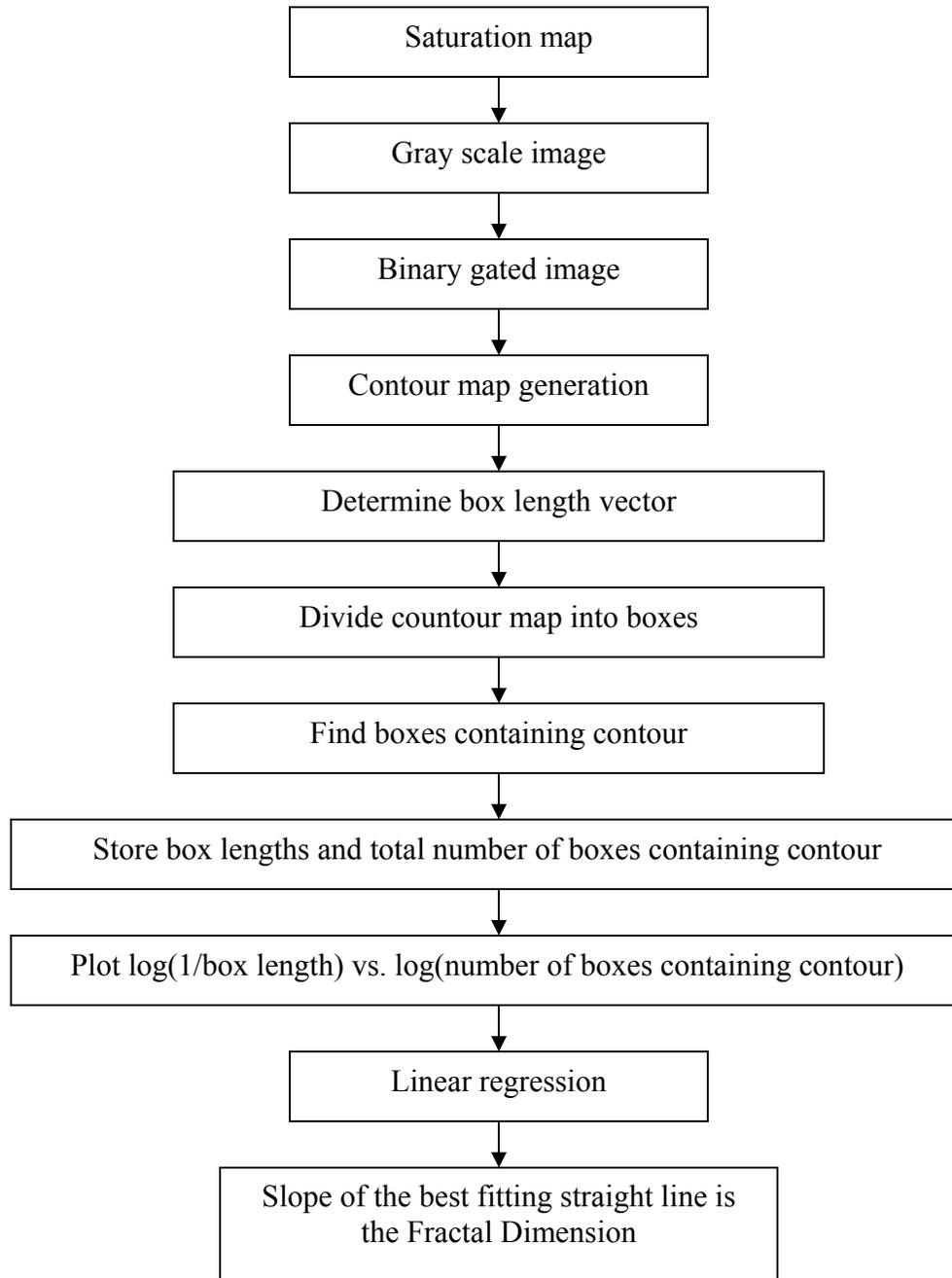


Figure 5: Flow chart of algorithm developed for determination fractal dimension using box counting technique

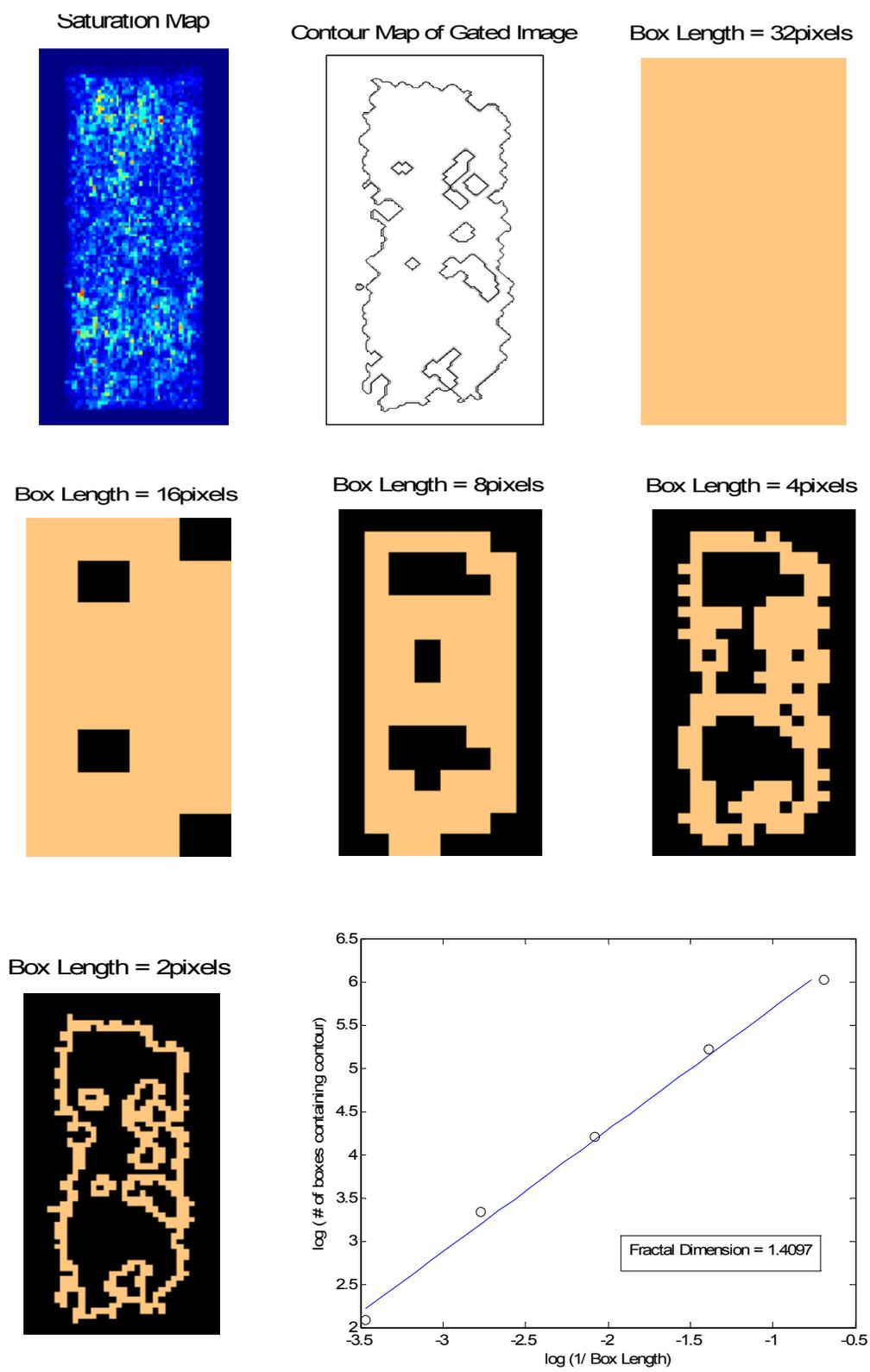


Figure 6: Determination of fractal dimension using box counting method

*Statistical Analysis:*

A quantitative comparison between the distributions of number of wet patches at different drying times obtained from the drying simulation and that from the NMR image was done by Kolmogorov-Smirnov statistics test.<sup>38</sup> Kolmogorov-Smirnov test statistic is the maximum of absolute differences between two cumulative distribution functions of the distributions being compared<sup>46</sup>. The change in fractal dimension of the saturated regions in case of simulation and NMR imaging experiment was compared with a p-value of 0.05.

## RESULTS AND DISCUSSIONS:

*NMR Imaging:*

The saturation or proton density maps obtained through both MRI drying experiment are shown in figure 6. Only four of the 16 proton density maps are shown here for clarity. The whole time series of proton density map is provided in the appendix section. The proton density maps correspond to the saturation distribution in the porous bed during drying. A clear drying front is not evident from these experimental data.

*Wet Patch Evolution:*

The spatial distribution of saturation within the sample is characterized, in this study, through the number of wet patches developed during the drying process. The evolution of the number of wet patches formed as the drying progressed are shown in figure 7, and 8 for 500 $\mu$  and 250 $\mu$  diameter beads, respectively. The y-error bars in the plots indicate the standard deviations among the three replications. A similar trend in the change in number of wet patches is observed in experimental and simulation data. The wet patch distribution data from experiment and simulation were compared quantitatively using Kolmogorov-Smirnov statistics. The value of Kolmogorov-Smirnov statistic in this case was 0.051, which is more

than 0.04702 the table value of  $w_{1-\alpha}$  quantile with the level of significance ( $\alpha$ ) as 0.05.

Hence, we can conclude that the distribution of wet patches during drying were statistically different (with significance level of 0.05) between experiment and simulation. This discrepancy can possibly be attributed to: imperfect binary gating of the MRImages leading to artificial rupture of some wet patches, and lower resolution of the imaging method leading to inaccurate assignment of pixels as part of wet or dry regions. More sophisticated segmentation techniques involving local statistical correlation, or adaptive computation based method can possibly improve the accuracy of wet patch detection.

#### *Fractal Dimension Analysis:*

As the spatial distribution of saturation during drying was observed to be inhomogeneous, the complexity of the structures of saturated regions was characterized quantitatively by their fractal dimensions. The fractal dimension of saturation map for both the sizes of beads took up values between 2 and 1 although two-dimensional images were analyzed. This is an expected outcome because of the complexity of the structure of the pore space. In both 250 $\mu$  and 500 $\mu$  beads the fractal numbers decreased as the drying progressed during the experiment. This experimental data agreed well with the simulation outputs as shown in figures 6, and 7. Surface treated glass beads used in this work being neither hydrophilic nor hydrophobic larger pores were expected to empty first before the smaller ones. As the drying progressed, saturated smaller pores became predominant constituent of the wet patches, reducing the overall fractal dimension. A statistical comparison between the time-series data of fractal dimensions in case of experimental data and simulation showed that they were equivalent at the p-value of 0.05.

#### *Drying Curve Analysis:*

The overall effect of drying of the glass bead packs were studied by drying curves, where the total saturation of the pack is recorded at regular time intervals. The comparisons between the gravimetric and simulation results are shown in figure 11 and 12 for 500 $\mu$  and 250 $\mu$  diameter beads, respectively. It is observed in the gravimetric data that the drying rate starts to reduce towards the end of drying. This pattern is not observed in the simulation results. But if only the first part of the drying curves are compared, a good agreement is observed between the simulation and the experimental data, showing that the realistic pore network based drying simulation is capable of predicting isothermal drying to a reasonable extent. In the simulation, it is assumed that there was no surface interaction between water and glass beads. This assumption was based on the microscopic examination of the contact angle after the surface treatment. Although, the contact angle was observed to be 88° in the surface treated capillary, the surface treatment on all the beads may not be uniform. There may be some residual hydrophilicity on the glass bead surface even after the surface treatment. This can contribute to a very thin strongly bound layer of water over the glass bead surface. Removal of this water during drying will be difficult. Hence, the rate of drying reduced towards the end of drying when the proportion of bound water in the total saturation increased significantly. This phenomenon is also evidenced by the fact that the total saturation of the bead pack did not go to zero at the end of drying, since removal of strongly bound closest layer of water needed enough activation energy, which could not be provided by the isothermal room temperature drying set up of the experiment. The faster drying of the 500 $\mu$  bead pack as compared to the 250 $\mu$  can be due to lower capillary resistance and throat density. In this study, a very small packed bed was used for experimentation, the air supply was at 25°C, air was flowing around the packed bed, and the flow rate was moderate. Hence

it was reasonable to assume that there was no thermal gradient developing in the bed during the drying process. A moderate amount of thermal gradient formed due to evaporative cooling can also contribute to slower drying in gravimetric experiment.

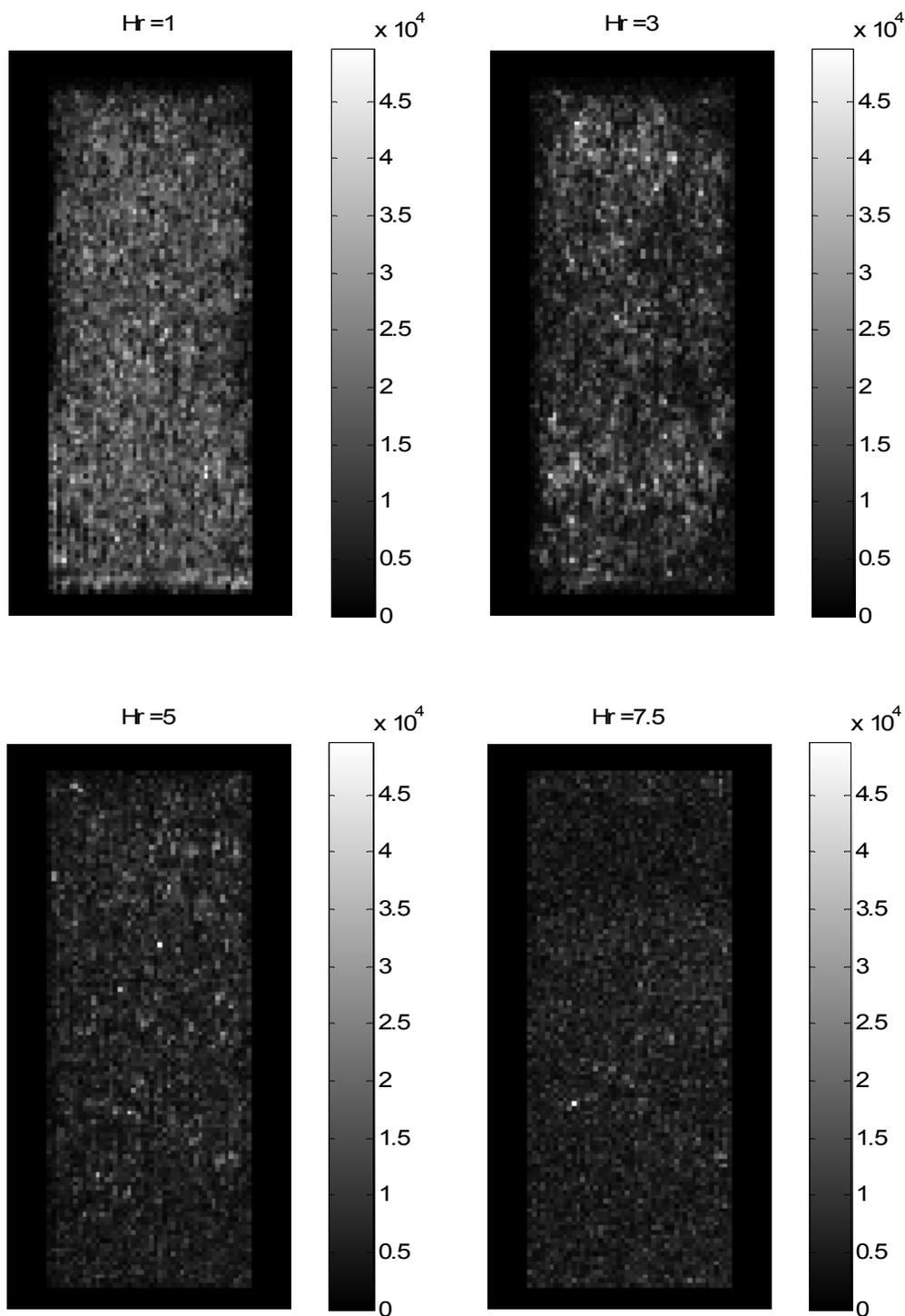


Figure 7: Proton density map of glass bead pack obtained through NMR spin-echo micro-imaging (all time points are shown in the appendix section).[sample dimension: 1cmx1cmx2cm; scales indicate water proton density]

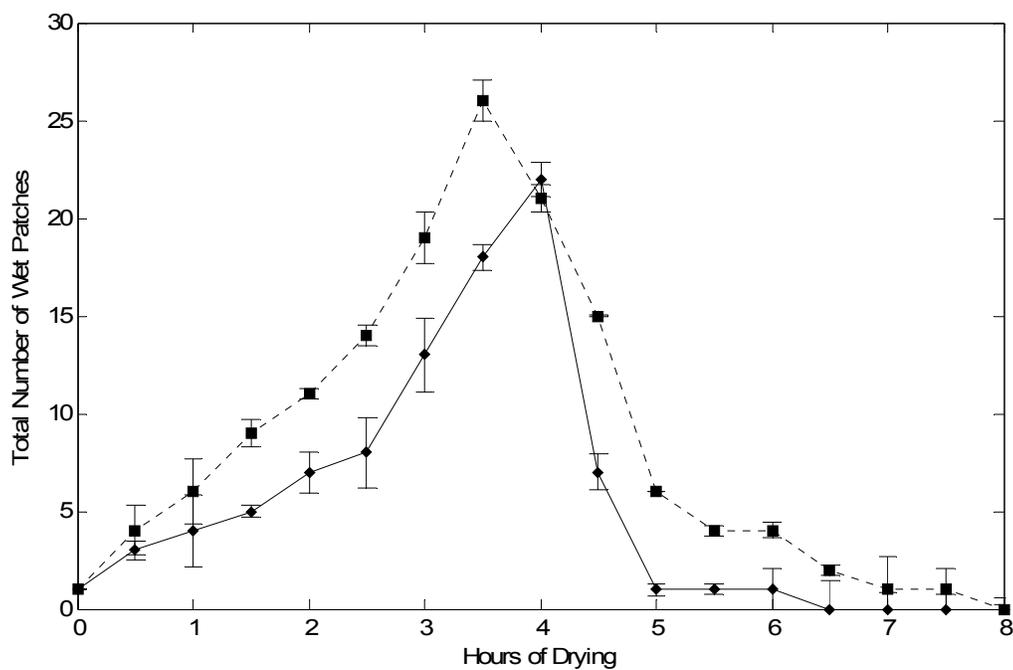


Figure 8: Evolution of wet patches during 500 $\mu$  mean diameter glass bead drying (dotted line experimental data, solid line simulation data)

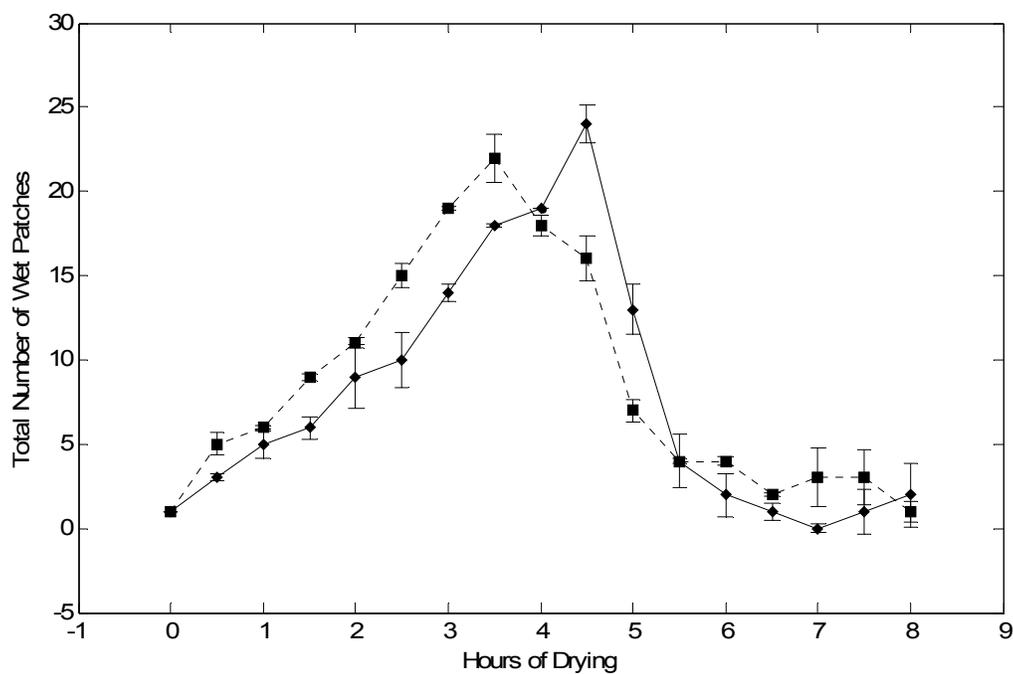


Figure 9: Evolution of wet patches during 250 $\mu$  mean diameter glass bead drying (dotted line experimental data, solid line simulation data)

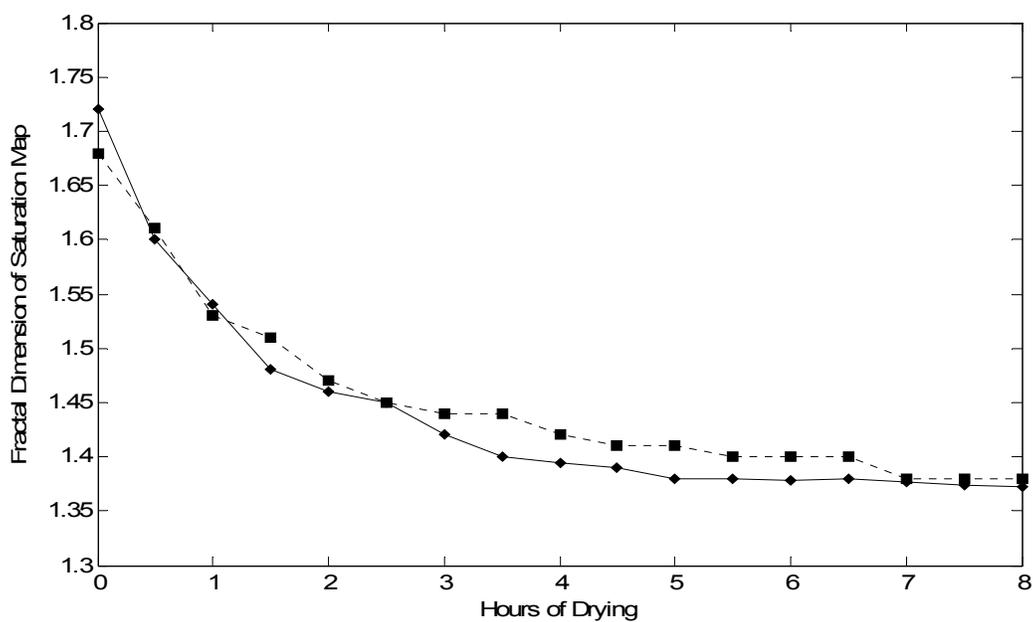


Figure 10: Change in fractal dimension during 500 $\mu$  mean diameter glass bead drying (dotted line experimental data, solid line simulation data)

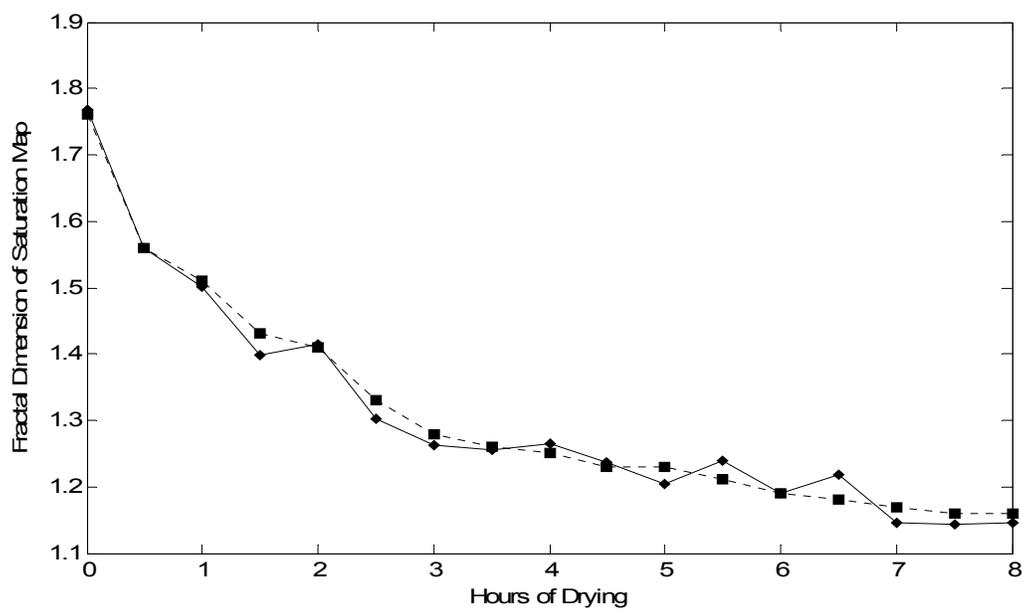


Figure 11: Change in fractal dimension during 250 $\mu$  mean diameter glass bead drying (dotted line experimental data, solid line simulation data)

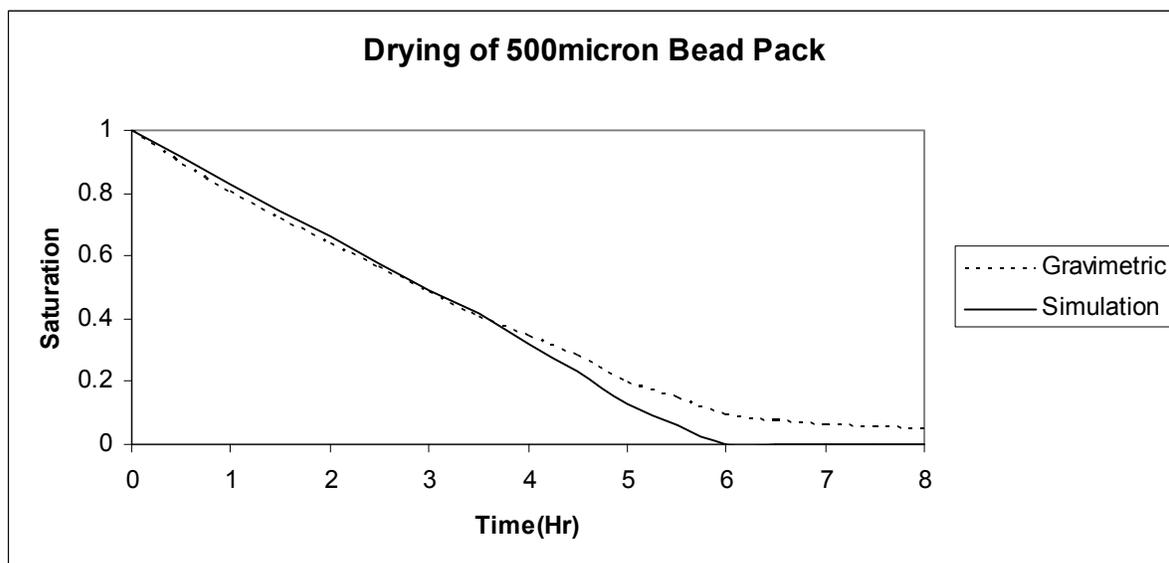


Figure 12: Drying curves from gravimetric measurement and simulation for 500 $\mu$  diameter beads

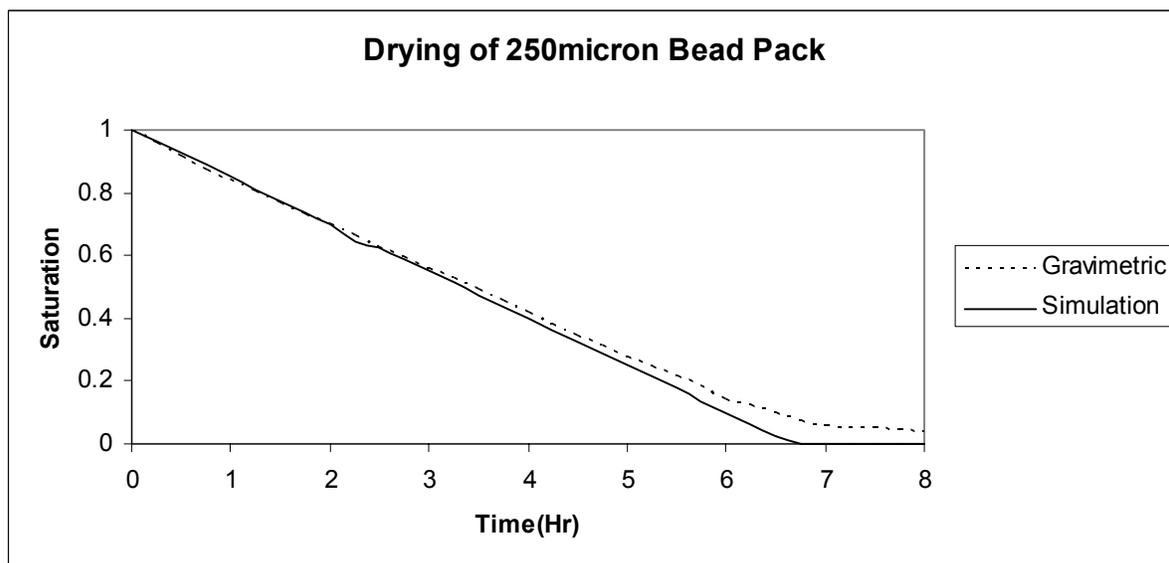


Figure 13: Drying curves from gravimetric measurement and simulation for 250 $\mu$  diameter beads

## CONCLUSION:

Drying of unconsolidated porous media under isothermal condition is investigated using non-invasive saturation mapping by in-situ magnetic resonance imaging and invasion percolation based simulation based on realistic pore network information extracted from reconstructed porous media. Quantitative characterization of experimental and simulation data was done by means of evolution of wet patches, fractal dimensions of saturated region, and drying curves. The wet patch evolution patterns were similar but statistically different. This discrepancy can be attributed to improper segmentation and assignment of pixels. The evolution of fractal dimension of saturated regions agreed well indicating that the pore network based method was able to simulate the spatial complexity of drying in porous media. The drying curves from gravimetric experiment and simulation agreed well in the higher saturation region. The difference in drying rate towards the end can be due to the surface effect of the glass beads.

## ACKNOWLEDGEMENT:

We acknowledge the help of the fabrication facility at Miami Valley Innovation Center, Cincinnati, OH for making the drying cell the current work.

## LITERATURE CITED:

1. Whitaker S. Simultaneous heat, mass and momentum transfer in porous media: a theory of drying. *Advances in Heat Transfer*. 1977;13:217-255.

2. Masmoudi W, Prat M, Bories S. Drying: percolation theory or continuum approach - some experimental evidences. In: Quiantard M, Todorovic M. Heat and Mass Transfer in Porous Media. New York: Elsevier, 1992:817-28.
3. Plumb OA, Prat M. Microscopic models for the study of drying of capillary porous media. In: Mujumdar AS. Drying '92. New York: Elsevier, 1992:397-406.
4. Laurindo JB, Prat M. Numerical and experimental network study of evaporation in capillary porous media. Phase distributions. Chem. Eng. Sci. 1996;51:5171-85.
5. Laurindo JB, Prat M. Numerical and experimental network study of evaporation in capillary porous media. Drying rates. Chem. Eng. Sci. 1998;53:2257-69.
6. Neimark AV. Multiscale percolation systems. Sov. Phys. JETP. 1989;69:786-91.
7. Bray YL, Prat M. Three-dimensional pore network simulation of drying in capillary porous media. Int. J. of Heat and Mass Transfer. 1999;42:4207-24.
8. Ghosh S, Keener KM, Bills MJ, Pan Y. Pore-Space Geometry Characterization of Simulated Unconsolidated Porous Media. AIChE J. (submitted for publication)
9. Gunstensen AK, Rothman DH. Lattice Boltzman studies of immiscible two-phase flow through porous media. J of Geophys Rev. 1993;98:6431-41.
10. Rothman DH, Zaleski S. Lattice-gas models of phase separation: interfaces, phase transitions, and multiphase flow. Reviews of Modern Physics. 1994;66:1417-79.
11. Dawson SP, Chen SY, Doolen GD, Janecky DR, Lawniczak A. Lattice methods and their applications to reacting systems. Computers and Chem Eng. 1995;19:617-46.
12. Blunt M, King P. Macroscopic parameters from simulations of pore scale flow. Physical Review A. 1990;42:4780-87.

13. Rajaram H, Ferrand LA, Celia MA. Prediction of relative permeabilities for unconsolidated soils using pore-scale network models. *Water Resources Research*. 1997;33:43–52.
14. Fischer U, Celia M. Prediction of relative and absolute permeabilities for gas and water retention curves using a pore-scale network model. *Water Resources Research*. 1999;35:1089–1100.
15. Jerauld GR, Salter SJ. The effect of pore-structure on hysteresis in relative permeability and capillary pressure: pore level modeling. *Transport in Porous Media*. 1990;5:103-51.
16. Bryant SP, King R, Mellor DW. Network model evaluation of permeability and spatial correlation in a real random sphere packing. *Transport in Porous Media*. 1993;11:53–70.
17. Reeves PC, Celia MA. A functional relationship between capillary pressure, saturation, and interfacial areas as revealed by a pore-scale network model. *Water Resources Research*. 1996;32:2345-58.
18. Thompson KE, Fogler HS. Modeling flow in disordered packed beds from pore-scale fluid mechanics. *AIChE J*. 1997;43:1377-89.
19. Lowry MI, Miller CT. Pore-scale modeling of nonwetting-phase residual in porous media. *Water Resources Research*. 1995;31:455-73.
20. Dillard LA, Blunt M. Development of a pore network simulation model to study nonaqueous phase liquid dissolution. *Water Resources Research*. 2000;36:439-54.
21. Dias MM, Payatakes AC. Network models for two-phase flow in porous media, 2, motion of oil ganglia. *Journal of Fluid Mechanics*. 1986;164:337-68.

22. Li Y, Wardlaw NC. Mechanisms of nonwetting phase trapping during imbibition at slow rates. *Journal of Colloid and Interface Sc.* 1986;109:461-472.
23. Li Y, Wardlaw NC. The influence of wettability and critical pore-throat size ratio on snap-off. *Journal of Colloid and Interface Sc.* 1986;109:473-486.
24. Satik C, Yortsos YC. A pore network study of bubble growth in porous media driven by heat transfer. *J. Heat Transf. Trans. ASME.* 1996;118:455-462.
25. McCall KR, Guyer RA. Fluid configurations in partially saturated porous media. *Phys. Rev. B.* 1991;43:808-815.
26. Prat M. Isothermal drying of non-hygroscopic capillary-porous materials as an invasion percolation process. *Int. J. Mult. Flow.* 1995;21:875-92.
27. Prat M. Percolation model of drying under isothermal conditions in porous media. *Int. J. Mult. Flow.* 1993;19:691-704.
28. Prat M, Bouleux F. Drying of capillary porous media with stabilized front in two-dimensions. *Phys. Rev. E.* 1999;60:5647-56.
29. Broadbent SR, Hammersley JM. The mathematics of percolation in networks. *Proc. Cambridge Philosophical Society.* 1957;53:629-35.
30. Chandler R, Koplik J, Larman K, Willemsen J. Capillary displacement and percolation in porous media. *J Fluid Mech.* 1982;119:249-53.
31. Wilkinson D, Willemsen JF. Invasion percolation: a new form of percolation theory. *J. Phys. A.* 1983;16:3365-76.
32. Wilkinson D. Percolation model of immiscible displacement in the presence of buoyancy forces. *Phys. Rev. A* 1984;30:520-31.

33. Glass RJ, Yarrington L. Simulation of gravity fingering in porous media using a modified invasion percolation model. 1996;70:231-252.
34. Shaw TM. Drying as an immiscible displacement process with fluid counterflow. Phys. Rev. Lett. 1987;59:1671-74.
35. Otsu N. A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics. 1979;9:62-66.
36. Foroutan-pour K, Dutilleul P, Smith DL. Advances in the implementation of the box-counting method of fractal dimension estimation. Applied Mathematics and Computation. 1999;105:195-210.
37. Mandelbrot BB. The Fractal Geometry of Nature (1st edition). New York: W. H. Freeman, 1982.
38. Conover WJ. Practical Nonparametric Statistics(1st edition). New York: Wiley, 1998.

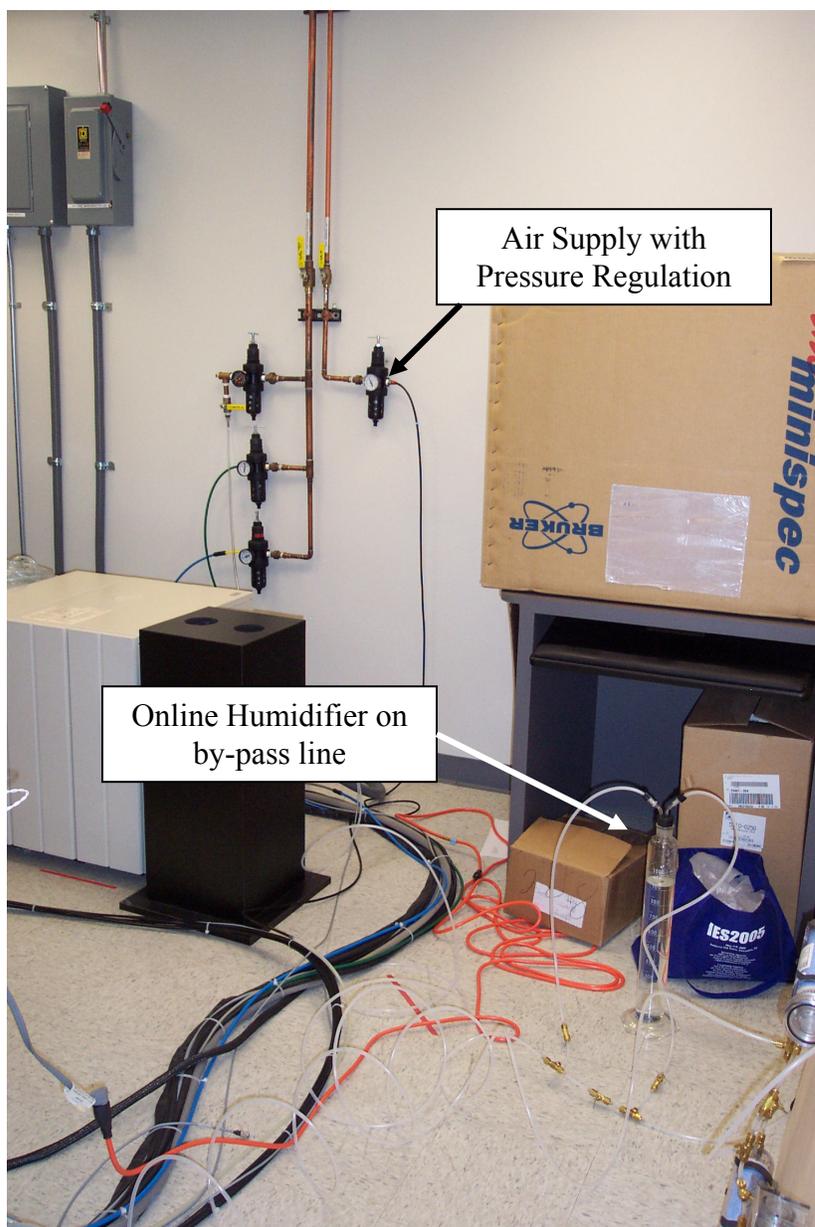


Figure 14: Set up for in-situ drying inside MRI instrument

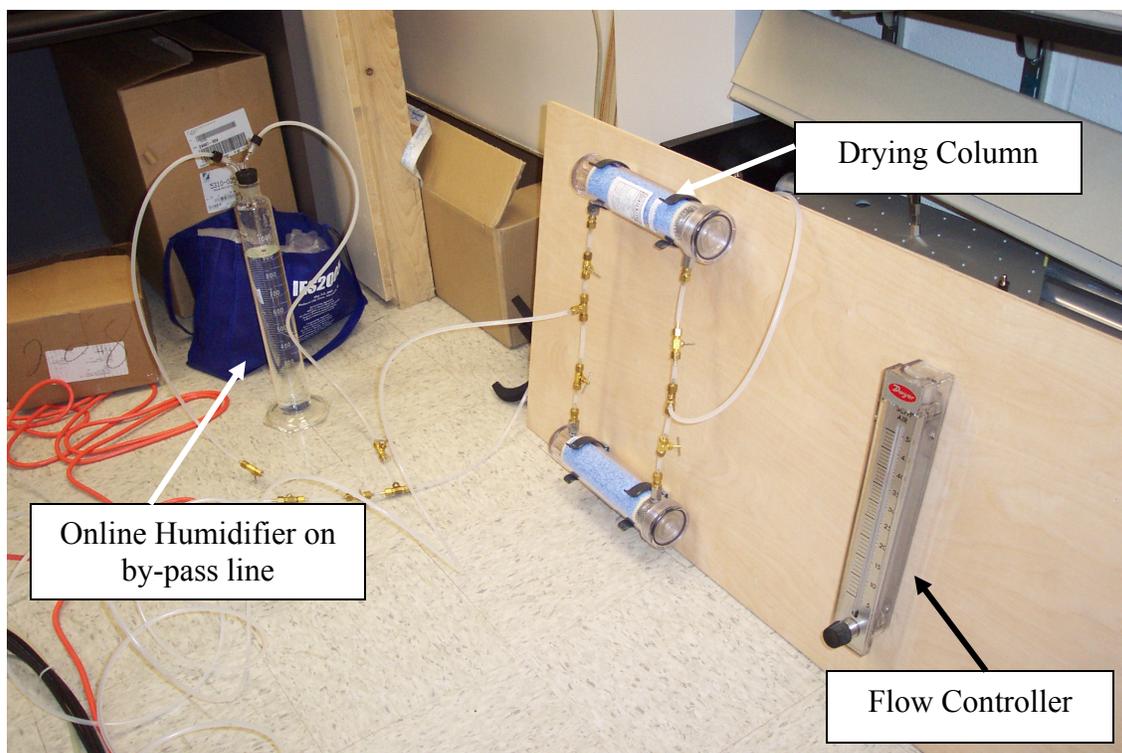


Figure 15: Drying columns, bypass line through humidifier, flow meter/controller

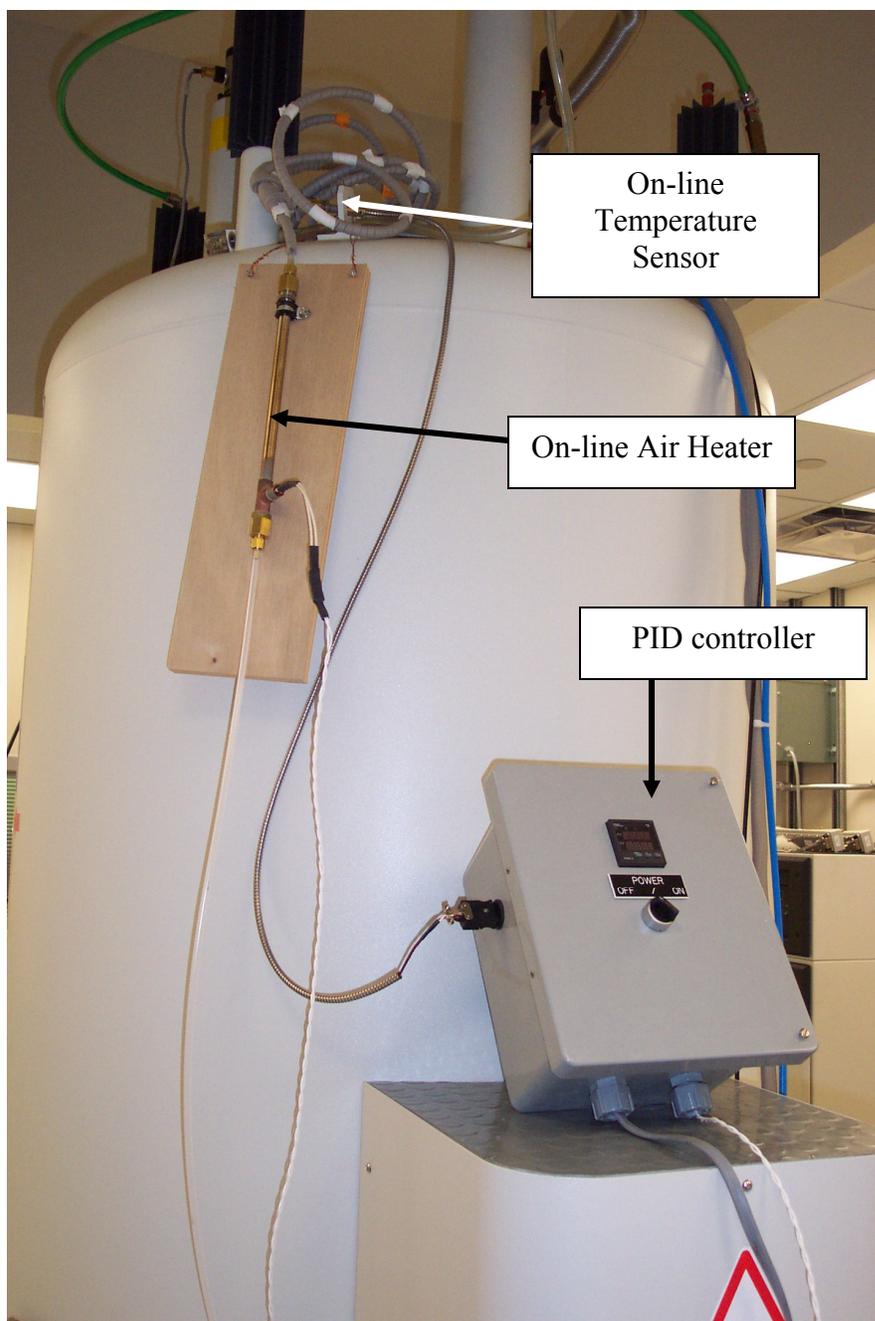


Figure 16: PID controller, heat exchanger, online temperature sensor, Bruker Ultrashield 500MHz microimaging system

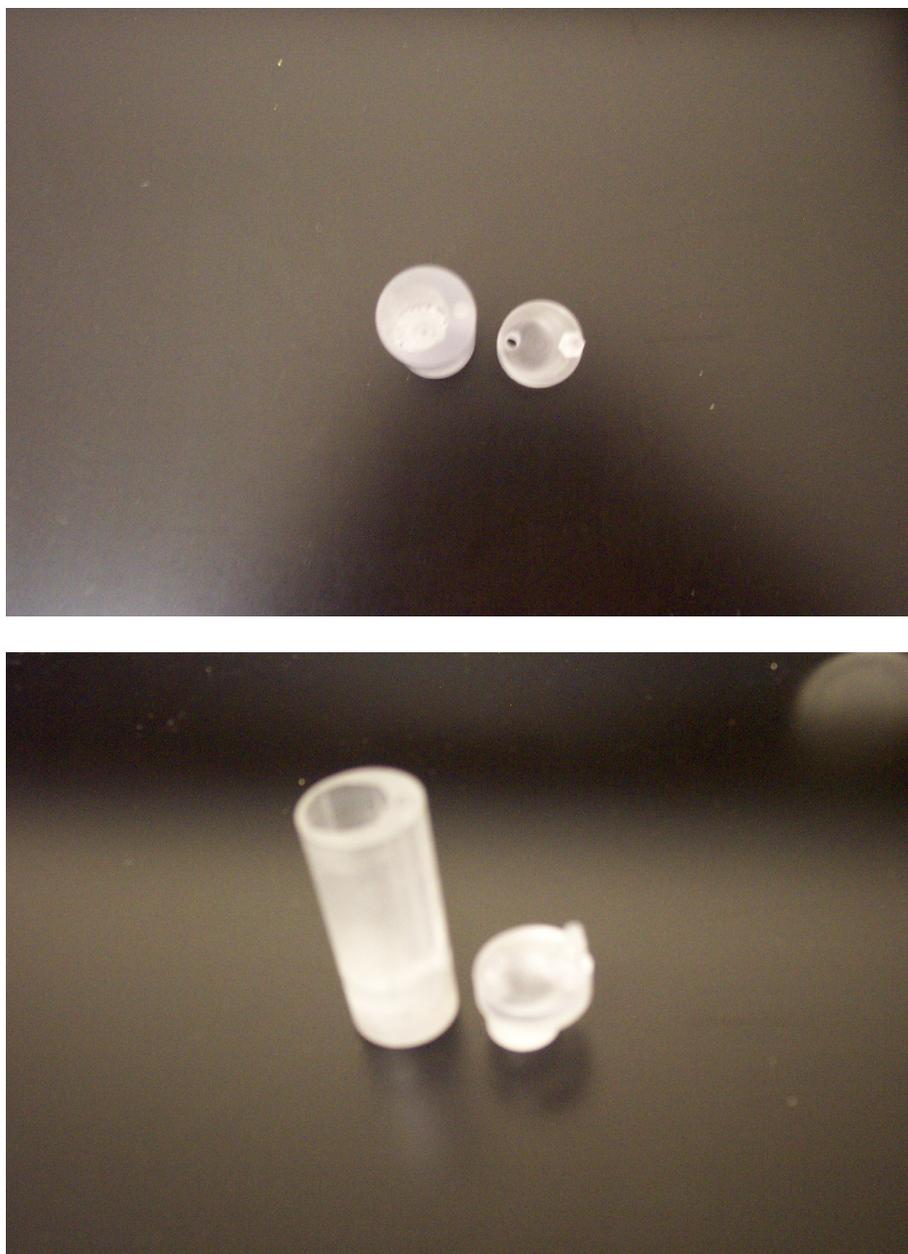


Figure 17: Drying Cell (a) top view (top image); (b) side view (bottom image)

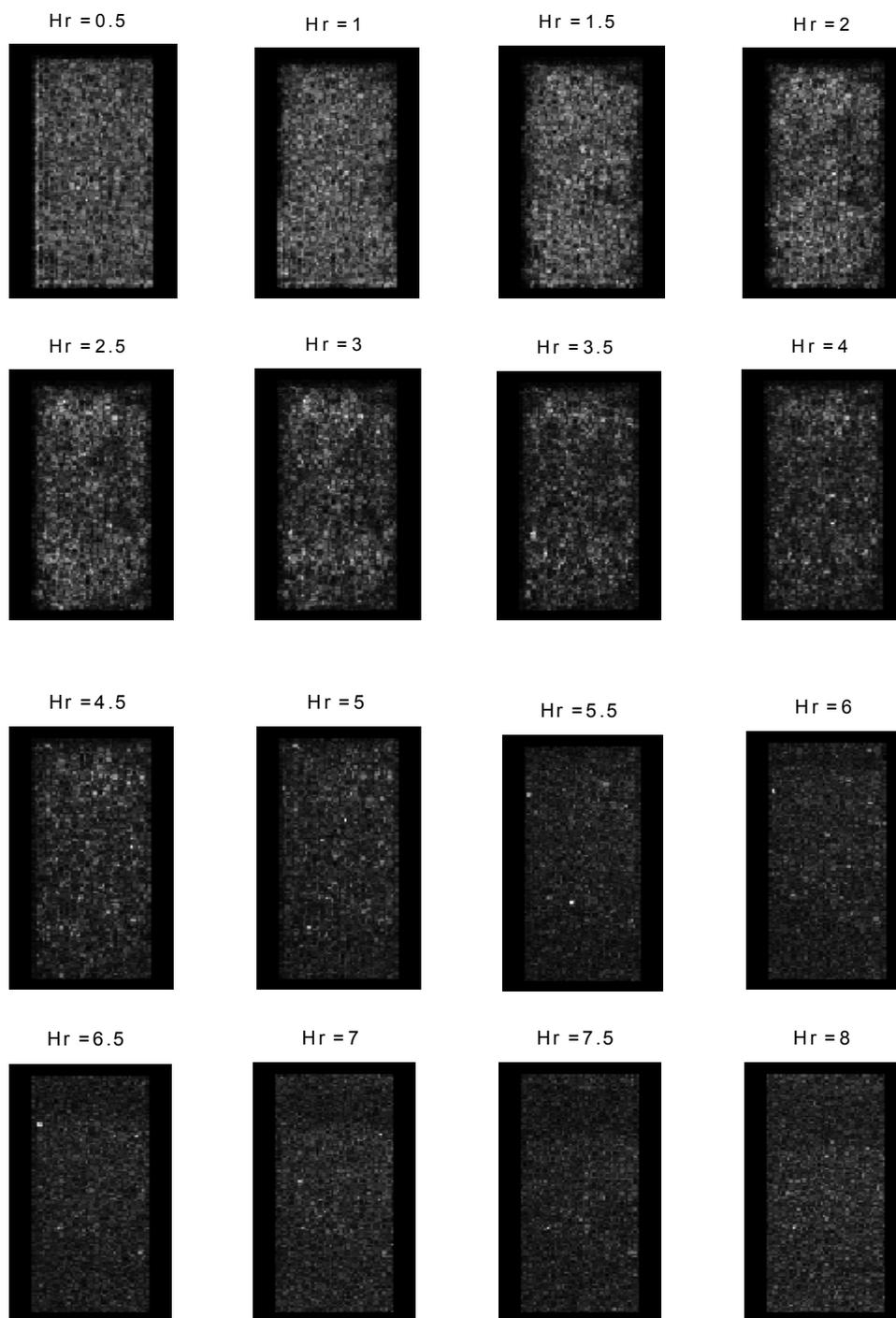


Figure 18: Proton density map of glass bead pack obtained through NMR spin-echo micro-imaging

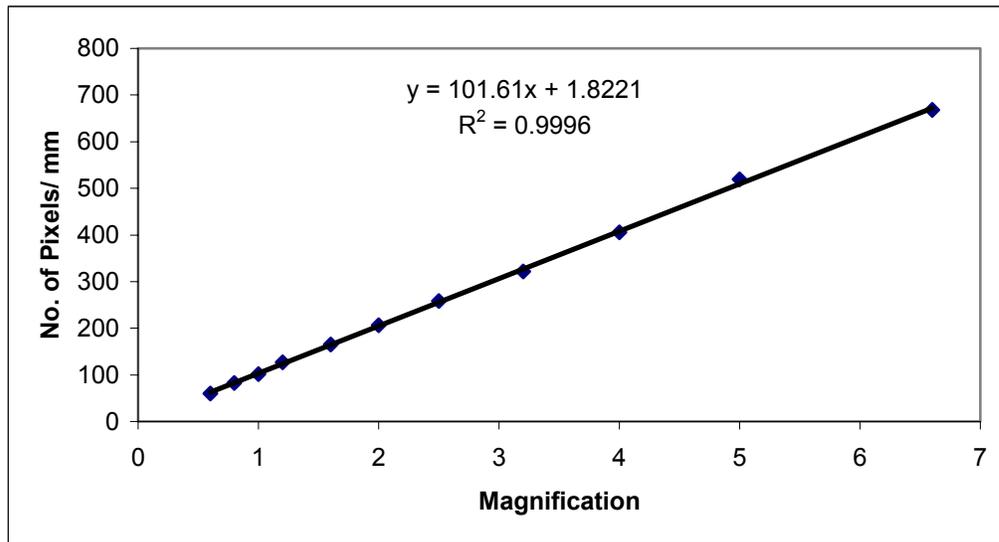


Figure 19: Calibration of Microscopic Images

Matlab codes for microscopic image analysis for automated bead size determination:

```
function [] = particle_size_distn()
%J=imread('C:\250_mag12_reg2.tif');
%imshow(J), title('Original Image')
I = imread('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\PhDExperiment\BeadSizeAnalysis\Images\Beads\500u_bottomlight\500u_reg20.tif');
%imshow(I), title('Original Image')
I = rgb2gray(I);
%figure, imshow(I), title('GrayLevel Image')
%I2 = medfilt2(I,[15 15]);
%figure, imshow(I2)
%background = imopen(I, strel('disk', 18));
%figure, imshow(background), title('background')
%I = imsubtract(I, background);
%figure, imshow(I), title('background removed')
%lev = graythresh(I)
%figure, imhist(I)
I = im2bw(I,0.70);
figure, imshow(I), title('Thresholded Image')
%I = imopen(I, strel('disk', 2));
%figure, imshow(I)
%se = strel('disk',5);
I = imfill(I,'holes');
figure, imshow(I), title('Filled Image')
L = bwlabel(I);
num_beads = max(L(:))
stats = regionprops(L, 'EquivDiameter');
allDia = [stats.EquivDiameter];
allDia = (allDia)
%figure, hist(allDia)
%MeanDia = mean(allDia)
%MedianDia = median(allDia)
%SD = std(allDia)
%MaxDia = max(allDia)
%MinDia = min(allDia)

Matlab codes for simulations
(a) 3D Monte Carlo simulation of particle packing
function [x_p,y_p,z_p,r_p]= sphere_mc_3d()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%=% input variables : none
% output variables : none
% subfunctions : checkoverlap, removeoverlap, changeposition
```

```

% author : Supriyo Ghosh
% last modified: Dec. 22, 2005
% This function does a MonteCarlo simulation of particles in a defined 3D space. The particle sizes
% are distributed normally. The 3D space has hard walls. Particles are hard particles.
% =====
%randomly generating particle sizes
m=25; %mean particle radius in microns
v=0; %particle radii variance
global xmax ymax zmax;
xmax=500; % dimension of container in x-direction
ymax=500; % dimension of container in x-direction
zmax=1500; % dimension of container in z-direction
pnum=150; % number of particles used
count=0;
e_potential_temp=10e100; %initializing total potential energy of the system
e_potential_save=[];
%=====
particles.r(1:pnum)=round(m+sqrt(v)*randn(1, pnnum)); %randomly assigning radii to all the particles
r=particles.r;
bx=xmax-r;
by=ymax-r;
bz=zmax-r;
particles.x(1:pnum)=r+(bx-r).*rand(1, pnnum); %generates random numbers in the range r:xmax-r
particles.y(1:pnum)=r+(by-r).*rand(1, pnnum); %generates random numbers in the range r:ymax-r
particles.z(1:pnum)=r+(bz-r).*rand(1, pnnum); %generates random numbers in the range r:zmax-r
for i=1:pnum
    p.x = particles.x(i);
    p.y = particles.y(i);
    p.z = particles.z(i);
    p.r = particles.r(i);
    e_interaction_sum = checkoverlap(particles,p,i);
    p = removeoverlap(particles,p,e_interaction_sum,i);
    particles.x(i) = p.x;
    particles.y(i) = p.y;
    particles.z(i) = p.z;
    particles.r(i) = p.r;
end
%=====initial position plot=====
figure,scatter3(particles.x, particles.y, particles.z), title('Initial Distribution')
% figure
% for k=1:pnum
% circle([particles.x(k) particles.z(k)],r(k),1);
% axis([0 xmax 0 zmax]);
% end
%=====
fid = fopen('TotalPotential','w');
while (count<=100) % number of iterations
    for i=1:pnum
        p.x = particles.x(i);
        p.y = particles.y(i);
        p.z = particles.z(i);
        p.r = particles.r(i);
        p = changeposition(p);
        particles.x(i) = p.x;
        particles.y(i) = p.y;
        particles.z(i) = p.z;
        particles.r(i) = p.r;
        e_interaction_sum = checkoverlap(particles,p,i);
        p = removeoverlap(particles,p,e_interaction_sum,i);
        particles.x(i) = p.x;
        particles.y(i) = p.y;
        particles.z(i) = p.z;
        particles.r(i) = p.r;
    end
%=====
% if (count==0)
%     for k=1:pnum
%         circle([particles.x(k) particles.z(k)],r(k),1);
%         axis([0 xmax 0 zmax]);
%     end
% end
%=====
e_potential=(r.^3).*particles.z; %calculate potential energy
e_potential=sum(e_potential); %of the whole system
fprintf(fid,'%10.2f\n',e_potential);
%=====save particle positions after every 50 iterations=====
R = rem(count,50);
if R == 0
    save(strcat('position',int2str(count)), 'particles');
end
count = count + 1;
count
end % while
fclose(fid);

for i=1:pnum
    x_p(i) = particles.x(i);
    y_p(i) = particles.y(i);
    z_p(i) = particles.z(i);
    r_p(i) = particles.r(i);
end
x_p = x_p';

```

```

y_p = y_p';
z_p = z_p';
r_p = r_p';

%=== final position plot=====
figure,scatter3(x_p,y_p,z_p),title('Final Distribution')
% figure,plot3(x_p,y_p,z_p,'r'),title('Final Distribution')
% figure
% for k=1:pnum
% circle([particles.x(k) particles.z(k)],r(k),1);
% axis([0 xmax 0 zmax]);
% end
%=====
%%%%%%%%%%
function [e_interaction_sum]=checkoverlap(particles,p,index)

% checks if particle p overlaps with any of neighborhood particle, neighbourhood radius defined as radius(p)+radius(largest particle)

e_interaction_sum=0; % initialize to zero each time

% check overlap between neighborhood particles
rmax = max(particles.r);
neighborhood_r = p.r + rmax; % maximum distance between two particles which are just touching is sum of current particle radius and maximum radius in the distribution
% select particles falling in this neighborhood
x1=p.x-neighborhood_r;
x2=p.x+neighborhood_r;
y1=p.y-neighborhood_r;
y2=p.y+neighborhood_r;
z1=p.z-neighborhood_r;
z2=p.z+neighborhood_r;

temp=(find(particles.x>(x1)& particles.x<(x2)& particles.y>(y1)& particles.y<(y2) & particles.z>(z1)& particles.z<(z2)));
NeighborhoodParticles.x=particles.x(temp);
NeighborhoodParticles.y=particles.y(temp);
NeighborhoodParticles.z=particles.z(temp);
NeighborhoodParticles.r=particles.r(temp);
pnum = length(NeighborhoodParticles.r);

for i=1:pnum
    pix=NeighborhoodParticles.x(i);
    piy=NeighborhoodParticles.y(i);
    piz=NeighborhoodParticles.z(i);
    s=sqrt((pix-p.x)^2+(piy-p.y)^2+(piz-p.z)^2);
    d=NeighborhoodParticles.r(i)+p.r;
    %calculating interaction energy=====
    if (s>=d | s==0)
        e_interaction=0; % no overlap
    else
        e_interaction=10e7;
        e_interaction_sum=e_interaction_sum+1; %overlap
    end
end

%%%%%%%%%%
function [p]=removeoverlap(particles,p,e_interaction_sum,index)
global xmax ymax zmax;

bx=xmax-p.r;
by=ymax-p.r;
bz=zmax-p.r;
while (e_interaction_sum~=0)
    % change position of p
    p = changeposition(p);

    % check if this new position overlaps with any particle
    e_interaction_sum = checkoverlap(particles,p,index);
    % if yes then iterate
end

%%%%%%%%%%

function [p] = changeposition(p)

global xmax ymax zmax;
bx=xmax-p.r;
by=ymax-p.r;
bz=zmax-p.r;
delta=10;
inc_x=(-1)+(2)*rand(1); % x increment = a random number between +delta and -delta
p.x=p.x + delta*inc_x;
if p.x>bx % if new position comes outside the right wall assign equal to wall
    p.x = bx;
end
if p.x<p.r % if new position comes outside the left wall assign equal to wall
    p.x = p.r;
end
inc_y=(-1)+(2)*rand(1); % y increment = a random number between +delta and -delta
p.y=p.y + delta*inc_y;
if p.y>by % if new position comes outside the front wall assign equal to wall

```



```

Image = zeros(100,100,150);
for p=1:pnum
    l = zeros(100,100,150);
    l(x_p(p),y_p(p),z_p(p))=1;
    D = bwdist(l,'euclidean');
    I2 = zeros(100,100,150);
    for i=1:100
        for j=1:100
            for k=1:150
                if(D(i,j,k)<=r_p(p))
                    I2(i,j,k)=1;
                else
                    I2(i,j,k)=0;
                end
            end
        end
    end
    Image = Image + I2;
    p
end
for i=1:100
    for j=1:100
        for k=1:150
            if(Image(i,j,k)==0)
                Image(i,j,k)=0;
            else
                Image(i,j,k)=1;
            end
        end
    end
end
end

I = ~Image(21:80,21:80,21:80);
bulk_porosity = sum(I(:))/(60*60*60);
%I3 = imrotate(Image,-90,'bilinear');

%imshow(I3(:,10),[],'n')

%filename = '3dsphere_pack.vox';
%I3 = permute(I3,[2,3,1]);
%I3 = flipdim(I3,3);
%I3 = flipdim(I3,1);
%fid = fopen(filename,'wb');
%fwrite(fid,I3,'double');
%fclose(fid);
(d) Coordination number determination:
function [contact_no,av_con_no] = coordination_number()

% calculates the average number of touching spheres in a simulated papcked bed
x_p = load('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-01\x_p.txt');
y_p = load('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-01\y_p.txt');
z_p = load('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-01\z_p.txt');
r_p = load('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-01\r_p.txt');
pnum = length(r_p);
Image = zeros(100,100,150);
for p=1:pnum
    l = zeros(100,100,150);
    l(x_p(p),y_p(p),z_p(p))=1;
    D = bwdist(l,'euclidean');
    I2 = zeros(100,100,150);
    for i=1:100
        for j=1:100
            for k=1:150
                if(D(i,j,k)<=r_p(p))
                    I2(i,j,k)=p;
                else
                    I2(i,j,k)=0;
                end
            end
        end
    end
    Image = Image + I2;
    p
end
nI = Image;
contact_no = 0;
for i=2:99
    for j=2:99
        for k=2:149
            if(Image(i,j,k)==0)
                m = 0;
                if (nl(i-1,j-1,k+1)==0)&(nl(i-1,j-1,k+1)~=nl(i,j,k))
                    m = m+1;
                elseif (nl(i-1,j,k+1)==0)&(nl(i-1,j,k+1)~=nl(i,j,k))
                    m = m+1;
                elseif (nl(i-1,j+1,k+1)==0)&(nl(i-1,j+1,k+1)~=nl(i,j,k))
                    m = m+1;
                elseif (nl(i,j-1,k+1)==0)&(nl(i,j-1,k+1)~=nl(i,j,k))
                    m = m+1;
                elseif (nl(i,j,k+1)==0)&(nl(i,j,k+1)~=nl(i,j,k))
                    m = m+1;
                elseif (nl(i+1,j-1,k+1)==0)&(nl(i+1,j-1,k+1)~=nl(i,j,k))
                    m = m+1;
            end
        end
    end
end

```

```

elseif (nl(i+1,j,k+1)~=0)&(nl(i+1,j,k+1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j+1,k+1)~=0)&(nl(i+1,j+1,k+1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j-1,k)~=0)&(nl(i-1,j-1,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j,k)~=0)&(nl(i-1,j,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j+1,k)~=0)&(nl(i-1,j+1,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j-1,k)~=0)&(nl(i+1,j-1,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j-1,k-1)~=0)&(nl(i-1,j-1,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i,j-1,k)~=0)&(nl(i,j-1,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j,k-1)~=0)&(nl(i-1,j,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i-1,j+1,k-1)~=0)&(nl(i-1,j+1,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j,k)~=0)&(nl(i+1,j,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i,j-1,k-1)~=0)&(nl(i,j-1,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j+1,k)~=0)&(nl(i+1,j+1,k)~=nl(i,j,k))
    m = m+1;
elseif (nl(i,j,k-1)~=0)&(nl(i,j,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j-1,k-1)~=0)&(nl(i+1,j-1,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j,k-1)~=0)&(nl(i+1,j,k-1)~=nl(i,j,k))
    m = m+1;
elseif (nl(i+1,j+1,k-1)~=0)&(nl(i+1,j+1,k-1)~=nl(i,j,k))
    m = m+1;
end
end
if (m ~= 0)
    contact_no = contact_no + 1;
end
end
end
end
end

av_con_no = contact_no / pnum;
(e) Radial distribution function determination
function [Map10]= rad_distn_func(Image)

% generates radial distribution function for a packed bed
%Image = load('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-10\Image_var-01.mat');
Image = -Image(1:100,1:100,1:100);
space = zeros(100,100,100);
space(50,50,:)=1; % axis of the cylinder
D = bwdist(space,'euclidean');
r1 = (30:-10:10); % outer radius of the cylindrical band
r2 = (20:-10:0); % inner radius of the cylindrical band
M = zeros(100,100);
for r = 1:length(r1)
    cyl = zeros(100,100,100);
    for i=1:100
        for j=1:100
            for k=1:100
                if (D(i,j,k)<=r1(r))&(D(i,j,k)>r2(r))
                    cyl(i,j,k)=1;
                end
            end
        end
    end
    s = Image .* cyl;
    secn = sum(s(:));
    voxel_vol = sum(cyl(:));
    rad_porosity(r) = secn/voxel_vol;
    for i=1:100
        for j=1:100
            if (cyl(i,j,1) == 1)
                M(i,j)=rad_porosity(r);
            end
        end
    end
end
end
Map10=M;
save('C:\Documents and Settings\Kevin Keener\My Documents\SUPRIYO\Analyt_Symp05\ParticlePacking-SLR\SpherePack_SLR\var-10\Map10',Map10);
%plot(rad_porosity,'r'), title('Porosity Radial Distribution Function')
%figure, plot(secn,'m'), title('section')
%figure, plot(voxel_vol,'g'), title('voxel volume')
% x = (1:1:100);
% y = (1:1:100);
% [X,Y,Z] = meshgrid(x,y,M);
% surf(X,Y,Z)
% axis square
% shading interp
%pcolor(M)

```

```

%imshow(M,[],'n')
%colormap jet
%axis equal tight
%title('Variance 0.01')
(f) Connected Skeleton:
function [skel,NUM] = skel_connected(image)

% temp = int8(Image_2(36:85,36:85,36:85));
temp = int8(image);
% temp = smooth3(temp,'box',3);
[s1,s2,s3] = size(temp);
nl = int8(zeros(s1+20,s2+20,s3+20));
nl(11:s1+10,11:s2+10,11:s3+10) = temp;
[s1,s2,s3]=size(nl);

[L,NUM1] = bwlabeln(nl,26);

max1 = max(max(max(L)))

for i=1:s1
    for j=1:s2
        for k=1:s3
            if (L(i,j,k) > 1)
                nl(i,j,k) = 0;
            end
        end
    end
end

[L,NUM2] = bwlabeln(nl,26);

max2 = max(max(max(L)))

counter = 1;
layer = 0;
H = 1;
nl_prev = nl;

while (H ~= 0)

    skin = int8(zeros(s1,s2,s3));
    for i=2:(s1-1)
        for j=2:(s2-1)
            for k=2:(s3-1)
                if (nl(i,j,k)==1)
                    N26 = nl(i-1,j-1,k+1) + nl(i-1,j,k+1) + nl(i-1,j+1,k+1) + ...
                        nl(i,j-1,k+1) + nl(i,j,k+1) + nl(i,j+1,k+1) + ...
                        nl(i+1,j-1,k+1) + nl(i+1,j,k+1) + nl(i+1,j+1,k+1) + ...
                        nl(i-1,j-1,k) + nl(i-1,j,k) + nl(i-1,j+1,k) + ...
                        nl(i,j-1,k) + nl(i,j+1,k) + ...
                        nl(i+1,j-1,k) + nl(i+1,j,k) + nl(i+1,j+1,k) + ...
                        nl(i-1,j-1,k-1) + nl(i-1,j,k-1) + nl(i-1,j+1,k-1) + ...
                        nl(i,j-1,k-1) + nl(i,j,k-1) + nl(i,j+1,k-1) + ...
                        nl(i+1,j-1,k-1) + nl(i+1,j,k-1) + nl(i+1,j+1,k-1);
                    % N6 = nl(i,j,k+1) + nl(i-1,j,k) + nl(i,j-1,k) + nl(i,j+1,k) + nl(i+1,j,k) + nl(i,j,k-1);
                    if (N26<26)
                        skin(i,j,k) = 1;
                    end
                end
            end
        end
    end
    s = sum(sum(sum(skin)))

    counter = 0;
    for i=4:(s1-3)
        for j=4:(s2-3)
            for k=4:(s3-3)
                if (skin(i,j,k)==1)
                    N26 = nl(i-1,j-1,k+1) + nl(i-1,j,k+1) + nl(i-1,j+1,k+1) + ...
                        nl(i,j-1,k+1) + nl(i,j,k+1) + nl(i,j+1,k+1) + ...
                        nl(i+1,j-1,k+1) + nl(i+1,j,k+1) + nl(i+1,j+1,k+1) + ...
                        nl(i-1,j-1,k) + nl(i-1,j,k) + nl(i-1,j+1,k) + ...
                        nl(i,j-1,k) + nl(i,j+1,k) + ...
                        nl(i+1,j-1,k) + nl(i+1,j,k) + nl(i+1,j+1,k) + ...
                        nl(i-1,j-1,k-1) + nl(i-1,j,k-1) + nl(i-1,j+1,k-1) + ...
                        nl(i,j-1,k-1) + nl(i,j,k-1) + nl(i,j+1,k-1) + ...
                        nl(i+1,j-1,k-1) + nl(i+1,j,k-1) + nl(i+1,j+1,k-1);
                    if (N26 > 2)
                        temp = nl(i-10:i+10,j-10:j+10,k-10:k+10);
                        [L,NUM_prev] = bwlabeln(temp);
                        temp(11,11,11) = 0;
                        [L,NUM_current] = bwlabeln(temp);
                        if (NUM_prev == NUM_current)
                            nl(i,j,k) = 0;
                            counter = counter + 1;
                            R = rem(counter,1000);
                            if (R == 0)
                                counter
                            end
                        end
                    end
                end
            end
        end
    end
end

```

```

        end
    end
end

    end
end
end

H = (nl ~= nl_prev);
H = sum(sum(sum(H)));

nl_prev = nl;
layer = layer + 1

% save(strcat('home/ghosh/SUPRIYO/PackingResults/skeleton/layer',int2str(layer)), 'nl');

% figure,imshow(nl(:,:,25),[]), title(strcat('layer',int2str(layer)))

end

skel = int8(nl(11:(s1-10),11:(s2-10),11:(s3-10)));
[L,NUM]=bwlabeln(skel,26);
(g) Ma-Sonka Skeletonization:
unction [nl] = ma_sonka_skeletonization(Image)

%This code tries to find the 3D skeleton of the pore-space using the template
% based parallel thinning algorithm proposed by Ma & Sonka,Comp. Vision & Image
% Understanding, vol.64,no.3,pp420-433,1996
sz = size(Image);
I = zeros(sz(1)+10,(sz(2)+10),(sz(3)+10));
I(6:(sz(1)+5),6:(sz(2)+5),6:(sz(3)+5)) = Image;
s = size(I);
nl = I; % creating a new 3D copy which will be updated in each new iteration
counter = 0 %initializing the iteration progress counter
deletable = 1; % initializing 'deletable' to enter the first iteration
while (sum(deletable(:)) ~= 0)
% p isolates the voxels in object which have atleast one background voxel in 26-neighborhood
p = zeros(s(1),s(2),s(3));
for i=1:s(1)
    for j=1:s(2)
        for k=1:s(3)
            if (nl(i,j,k) ~= 0)
                conn_26 = nl(i-1,j-1,k+1) + nl(i-1,j,k+1) + nl(i-1,j+1,k+1) + nl(i,j-1,k+1) +...
                    nl(i,j,k+1) + nl(i+1,j-1,k+1) + nl(i+1,j,k+1) + nl(i+1,j+1,k+1) +...
                    nl(i-1,j-1,k) + nl(i-1,j,k) + nl(i-1,j+1,k) + nl(i+1,j-1,k) +...
                    nl(i-1,j-1,k-1) + nl(i,j-1,k) + nl(i-1,j,k-1) + nl(i-1,j+1,k-1) +...
                    nl(i+1,j,k) + nl(i,j-1,k-1) + nl(i+1,j+1,k) + nl(i,j,k-1) +...
                    nl(i+1,j-1,k-1) + nl(i+1,j,k-1) + nl(i+1,j+1,k-1);
                if (conn_26 < 26)
                    p(i,j,k) = 1;
                end
            end
        end
    end
end
end
end

% if p has only one non-zero(object) neighbor in the 26-connectivity neighbor i.e. a line-end-point, then it is not deletable
for i=1:s(1)
    for j=1:s(2)
        for k=1:s(3)
            if (p(i,j,k) ~= 0)
                conn_26 = nl(i-1,j-1,k+1) + nl(i-1,j,k+1) + nl(i-1,j+1,k+1) + nl(i,j-1,k+1) +...
                    nl(i,j,k+1) + nl(i+1,j-1,k+1) + nl(i+1,j,k+1) + nl(i+1,j+1,k+1) +...
                    nl(i-1,j-1,k) + nl(i-1,j,k) + nl(i-1,j+1,k) + nl(i+1,j-1,k) +...
                    nl(i-1,j-1,k-1) + nl(i,j-1,k) + nl(i-1,j,k-1) + nl(i-1,j+1,k-1) +...
                    nl(i+1,j,k) + nl(i,j-1,k-1) + nl(i+1,j+1,k) + nl(i,j,k-1) +...
                    nl(i+1,j-1,k-1) + nl(i+1,j,k-1) + nl(i+1,j+1,k-1);
                if (conn_26 == 1)
                    p(i,j,k) = 0;
                end
            end
        end
    end
end
end

% retain only non-tail points
for i=1:s(1)
    for j=1:s(2)
        for k=1:s(3)
            if (p(i,j,k) ~= 0)
                conn_26 = nl(i-1,j-1,k+1) + nl(i-1,j,k+1) + nl(i-1,j+1,k+1) + nl(i,j-1,k+1) +...
                    nl(i,j,k+1) + nl(i+1,j-1,k+1) + nl(i+1,j,k+1) + nl(i+1,j+1,k+1) +...
                    nl(i-1,j-1,k) + nl(i-1,j,k) + nl(i-1,j+1,k) + nl(i+1,j-1,k) +...
                    nl(i-1,j-1,k-1) + nl(i,j-1,k) + nl(i-1,j,k-1) + nl(i-1,j+1,k-1) +...
                    nl(i+1,j,k) + nl(i,j-1,k-1) + nl(i+1,j+1,k) + nl(i,j,k-1) +...
                    nl(i+1,j-1,k-1) + nl(i+1,j,k-1) + nl(i+1,j+1,k-1);
                if (conn_26 == 2) % if p has only 2 non-zero neighbors then check the following for non-tail point
                    if (nl(i,j,k-1)==1)&(nl(i,j+1,k)==1)&(nl(i-1,j,k)==1)
                        p(i,j,k) = 0;
                    elseif (nl(i,j,k-1)==1)&(nl(i-1,j,k)==1)&(nl(i,j+1,k)==1)
                        p(i,j,k) = 0; %either {s(p)& e(p)} or {s(p) & u(p)} are object points but not both
                    end
                end
            end
        end
    end
end

```

```

elseif (nl(i,j,k+1)==1)&(nl(i,j-1,k)==1)&(nl(i-1,j,k)==1)
    p(i,j,k) = 0;
elseif (nl(i-1,j,k)==1)&(nl(i,j-1,k)==1)&(nl(i,j,k+1)==1)
    p(i,j,k) = 0;%either {n(p)& w(p)} or {u(p) & w(p)} are object points but not both
elseif (nl(i,j,k+1)==1)&(nl(i+1,j,k)==1)&(nl(i,j+1,k)==1)
    p(i,j,k) = 0;
elseif (nl(i,j+1,k)==1)&(nl(i+1,j,k)==1)&(nl(i,j,k+1)==1)
    p(i,j,k) = 0;%either {n(p)& d(p)} or {e(p) & d(p)} are object points but not both
end
end
end
end
end
end
end

```

```

% matching the four classes(i.e. A, B, C, D) of templates
deletable = zeros(s(1),s(2),s(3));

```

```

% matching with class A

```

```

for i=1:s(1)
    for j=1:s(2)
        for k=1:s(3)
            if (p(i,j,k)~=0)
                % checking template A(a)
                if (nl(i,j+1,k)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i,j-1,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i-1,j-1,k)==0)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i,j-1,k-1)==0)&(nl(i+1,j-1,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template A(b)
                elseif (nl(i,j-1,k)==1)&(nl(i,j-2,k)==1)&(nl(i-1,j+1,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i+1,j+1,k+1)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j+1,k-1)==0)&(nl(i+1,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template A(c)
                elseif (nl(i,j,k+1)==1)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j-1,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i,j+1,k-1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template A(d)
                elseif (nl(i,j,k-1)==1)&(nl(i,j,k-2)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i+1,j+1,k+1)==0)
                    deletable(i,j,k) = 1;
                % checking template A(e)
                elseif (nl(i-1,j,k)==1)&(nl(i+1,j-1,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i+1,j+1,k+1)==0)&(nl(i+1,j-1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template A(f)
                elseif (nl(i+1,j,k)==1)&(nl(i+2,j,k)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
                end
            end
        end
    end
end
end

```

```

% matching with class B

```

```

for i=1:s(1)
    for j=1:s(2)
        for k=1:s(3)
            if (p(i,j,k)~=0)
                % checking template B(a)
                if (nl(i-1,j,k)==1)&(nl(i+1,j,k)==1)&(nl(i-1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(b)
                elseif (nl(i,j+1,k)==1)&(nl(i+1,j,k)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i,j-1,k+1)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i,j-1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(c)
                elseif (nl(i,j,k-1)==1)&(nl(i+1,j,k)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i,j-1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(d)
                elseif (nl(i,j,k+1)==1)&(nl(i+1,j,k)==1)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j-1,k-1)==0)&(nl(i,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(e)
                elseif (nl(i-1,j,k)==1)&(nl(i,j+1,k)==1)&(nl(i,j-1,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i,j-1,k-1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(f)
                elseif (nl(i-1,j,k)==1)&(nl(i,j,k+1)==1)&(nl(i+1,j-1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i,j-1,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i,j+1,k-1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(g)
                elseif (nl(i-1,j,k)==1)&(nl(i,j-1,k)==1)&(nl(i+1,j,k+1)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j,k-1)==0)
                    deletable(i,j,k) = 1;
                % checking template B(h)
                elseif (nl(i-1,j,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j-1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i+1,j+1,k+1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j+1,k-1)==0)
                    deletable(i,j,k) = 1;
            end
        end
    end
end

```

```

% checking template B(i)
elseif (nl(i,j,k+1)==1)&(nl(i,j+1,k)==1)&(nl(i-1,j-1,k)==0)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i,j-1,k-1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
% checking template B(j)
elseif (nl(i,j+1,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j,k-2)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i,j-1,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i-1,j-1,k)==0)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)
deletable(i,j,k) = 1;
% checking template B(k)
elseif (nl(i,j-1,k)==1)&(nl(i,j-2,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j,k-2)==1)&(nl(i-1,j,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i+1,j+1,k+1)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)
deletable(i,j,k) = 1;
% checking template B(l)
elseif (nl(i,j,k+1)==1)&(nl(i,j-1,k)==1)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j+1,k-1)==0)&(nl(i+1,j+1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
end
end
end
end
end

% matching with class C
for i=1:s(1)
for j=1:s(2)
for k=1:s(3)
if (p(i,j,k)~=0)
% checking template C(a)
if (nl(i,j-1,k)==1)&(nl(i+1,j,k)==1)&(nl(i,j,k-1)==1)&(nl(i-1,j,k+1)==0)&(nl(i-1,j+1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i,j+1,k+1)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)
deletable(i,j,k) = 1;
% checking template C(b)
elseif (nl(i,j,k+1)==1)&(nl(i,j-1,k)==1)&(nl(i+1,j,k)==1)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j+1,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
% checking template C(c)
elseif (nl(i,j,k+1)==1)&(nl(i,j+1,k)==1)&(nl(i-1,j,k)==1)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i,j-1,k)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i-1,j,k-1)==0)&(nl(i-1,j-1,k-1)==0)&(nl(i,j-1,k-1)==0)
deletable(i,j,k) = 1;
% checking template C(d)
elseif (nl(i,j+1,k)==1)&(nl(i+1,j,k)==1)&(nl(i,j,k-1)==1)&(nl(i-1,j-1,k+1)==0)&(nl(i-1,j,k+1)==0)&(nl(i,j-1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)
deletable(i,j,k) = 1;
% checking template C(e)
elseif (nl(i-1,j,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j-1,k)==1)&(nl(i,j-2,k)==1)&(nl(i-1,j,k+1)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)
deletable(i,j,k) = 1;
% checking template C(f)
elseif (nl(i,j,k+1)==1)&(nl(i-1,j,k)==1)&(nl(i,j-1,k)==1)&(nl(i,j-2,k)==1)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)&(nl(i,j,k-1)==0)&(nl(i,j+1,k-1)==0)&(nl(i+1,j,k-1)==0)&(nl(i+1,j+1,k-1)==0)
deletable(i,j,k) = 1;
% checking template C(g)
elseif (nl(i-1,j,k)==1)&(nl(i,j+1,k)==1)&(nl(i,j,k+1)==1)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i+1,j,k)==0)&(nl(i-1,j,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j-1,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
% checking template C(h)
elseif (nl(i-1,j,k)==1)&(nl(i,j+1,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j-1,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i+1,j-1,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i,j-1,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i+1,j,k)==0)
deletable(i,j,k) = 1;
% checking template C(i)
elseif (nl(i-1,j,k)==1)&(nl(i,j-1,k)==1)&(nl(i,j,k-1)==1)&(nl(i,j,k-2)==1)&(nl(i+1,j,k+1)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)
deletable(i,j,k) = 1;
end
end
end
end
end

% matching with class D
for i=1:s(1)
for j=1:s(2)
for k=1:s(3)
if (p(i,j,k)~=0)
% checking template D(a)
if (nl(i+1,j-1,k)==1)&(nl(i,j-1,k)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)
deletable(i,j,k) = 1;
% checking template D(b)
elseif (nl(i-1,j-1,k)==1)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i,j-1,k)==0)
deletable(i,j,k) = 1;
% checking template D(c)
elseif (nl(i-1,j+1,k)==1)&(nl(i,j+1,k)==0)&(nl(i+1,j+1,k)==0)&(nl(i+1,j,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i,j-1,k)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)
deletable(i,j,k) = 1;
% checking template D(d)
elseif (nl(i+1,j-1,k)==1)&(nl(i+1,j,k)==0)&(nl(i+1,j-1,k)==0)&(nl(i,j-1,k)==0)&(nl(i-1,j-1,k)==0)&(nl(i-1,j,k)==0)&(nl(i-1,j+1,k)==0)&(nl(i,j+1,k)==0)
deletable(i,j,k) = 1;
% checking template D(e)
elseif (nl(i-1,j,k+1)==1)&(nl(i,j,k+1)==0)&(nl(i+1,j,k+1)==0)&(nl(i-1,j,k)==0)&(nl(i+1,j,k)==0)&(nl(i-1,j,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
% checking template D(f)
elseif (nl(i+1,j,k+1)==1)&(nl(i-1,j,k+1)==0)&(nl(i,j,k+1)==0)&(nl(i-1,j,k)==0)&(nl(i+1,j,k)==0)&(nl(i-1,j,k-1)==0)&(nl(i,j,k-1)==0)&(nl(i+1,j,k-1)==0)
deletable(i,j,k) = 1;
% checking template D(g)

```



```

        intersect(i,j) = 1;
    % 14th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 15th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 16th type of possible intersection
    elseif (sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 17th type of possible intersection
    elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 18th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 19th type of possible intersection
    elseif (sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    end
end
end
D1_sk(:,k) = intersect;
end
% repairing discontinuities in direction perpendicular to the slices
% temp = zeros(s1, s2, s3);
% for i=2:(s1-1)
%     for j=2:(s2-1)
%         for k=2:(s3-1)
%             if (D1_sk(i,j,k)==0)
%                 if (D1_sk(i,j,k-1)==1)&(D1_sk(i,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i-1,j-1,k-1)==1)&(D1_sk(i+1,j+1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i-1,j,k-1)==1)&(D1_sk(i+1,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i-1,j+1,k-1)==1)&(D1_sk(i+1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i,j+1,k-1)==1)&(D1_sk(i,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i+1,j+1,k-1)==1)&(D1_sk(i-1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i+1,j,k-1)==1)&(D1_sk(i-1,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i+1,j-1,k-1)==1)&(D1_sk(i-1,j+1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D1_sk(i,j-1,k-1)==1)&(D1_sk(i,j+1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%             end
%         end
%     end
% end
% D1_sk = D1_sk + temp;

```

TB = 1

%Skeletonization and intersection finding in "Front-Back" direction

```

D2_sk = zeros(s1, s2, s3);
for k=1:s1
    page = Image(k,:);
    page = reshape(page,s2,s3);
    sk = bwmorph(page,'skel','Inf');
    sk = double(sk);
    %checking for intersection points
    intersect = zeros(s2,s3);
    for i=2:(s2-1)
        for j=2:(s3-1)
            if (sk(i,j)==1)
                % 1st type of possible intersection
                if (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 2nd type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 3rd type of possible intersection
                elseif (sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 4th type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 5th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
            end
        end
    end
end

```

```

    intersect(i,j) = 1;
    % 6th type of possible intersection
    elseif (sk(i-1,j)==1)&(sk(i,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 7th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 8th type of possible intersection
    elseif (sk(i-1,j+1)==1)&(sk(i,j-1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 9th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 10th type of possible intersection
    elseif (sk(i-1,j+1)==1)&(sk(i,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 11th type of possible intersection
    elseif (sk(i-1,j)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 12th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 13th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 14th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 15th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 16th type of possible intersection
    elseif (sk(i-1,j+1)==1)&(sk(i,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 17th type of possible intersection
    elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
        intersect(i,j) = 1;
    % 18th type of possible intersection
    elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    % 19th type of possible intersection
    elseif (sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
        intersect(i,j) = 1;
    end
end
end
end
D2_sk(k,:,:) = intersect;
end

% repairing discontinuities in direction perpendicular to the slices
% temp = zeros(s1, s2, s3);
% for i=2:(s1-1)
%     for j=2:(s2-1)
%         for k=2:(s3-1)
%             if (D2_sk(i,j,k)==0)
%                 if (D2_sk(i-1,j-1,k-1)==1)&(D2_sk(i+1,j+1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i-1,j,k-1)==1)&(D2_sk(i+1,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i-1,j+1,k-1)==1)&(D2_sk(i+1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i,j-1,k-1)==1)&(D2_sk(i,j+1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i,j,k-1)==1)&(D2_sk(i,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i,j+1,k-1)==1)&(D2_sk(i,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i+1,j-1,k-1)==1)&(D2_sk(i-1,j+1,k-1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i+1,j,k-1)==1)&(D2_sk(i-1,j,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D2_sk(i+1,j+1,k-1)==1)&(D2_sk(i-1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%             end
%         end
%     end
% end
% end
D2_sk = D2_sk + temp;

FB=1

%Skeletonization and intersection finding in "Left-Right" direction
D3_sk = zeros(s1, s2, s3);

```

```

for k=1:s2
    page = Image(:,k,:);
    page = reshape(page,s1,s3);
    sk = bwmorph(page,'skel',Inf);
    sk = double(sk);
    %checking for intersection points
    intersect = zeros(s1,s3);
    for i=2:(s1-1)
        for j=2:(s3-1)
            if (sk(i,j)==1)
                % 1st type of possible intersection
                if (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 2nd type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 3rd type of possible intersection
                elseif (sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 4th type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 5th type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 6th type of possible intersection
                elseif (sk(i-1,j+1)==1)&(sk(i,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 7th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 8th type of possible intersection
                elseif (sk(i-1,j+1)==1)&(sk(i,j-1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 9th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j)==0)&(sk(i-1,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 10th type of possible intersection
                elseif (sk(i-1,j+1)==1)&(sk(i,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 11th type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
                    intersect(i,j) = 1;
                % 12th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 13th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
                    intersect(i,j) = 1;
                % 14th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i-1,j+1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 15th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 16th type of possible intersection
                elseif (sk(i-1,j+1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
                    intersect(i,j) = 1;
                % 17th type of possible intersection
                elseif (sk(i-1,j)==1)&(sk(i,j-1)==1)&(sk(i,j+1)==1)&(sk(i+1,j)==1)&(sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)
                    intersect(i,j) = 1;
                % 18th type of possible intersection
                elseif (sk(i-1,j-1)==1)&(sk(i+1,j-1)==1)&(sk(i+1,j+1)==1)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j-1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
                    intersect(i,j) = 1;
                % 19th type of possible intersection
                elseif (sk(i-1,j-1)==0)&(sk(i-1,j+1)==0)&(sk(i+1,j-1)==0)&(sk(i+1,j+1)==0)&(sk(i-1,j)==0)&(sk(i-1,j+1)==0)&(sk(i,j+1)==0)&(sk(i+1,j)==0)
                    intersect(i,j) = 1;
                end
            end
        end
    end
    D3_sk(:,k,:) = intersect;
end

% repairing discontinuities in direction perpendicular to the slices
% temp = zeros(s1, s2, s3);
% for i=2:(s1-1)
%     for j=2:(s2-1)
%         for k=2:(s3-1)
%             if (D3_sk(i,j,k)==0)
%                 if (D3_sk(i-1,j+1,k-1)==1)&(D3_sk(i+1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D3_sk(i,j+1,k-1)==1)&(D3_sk(i,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D3_sk(i+1,j+1,k-1)==1)&(D3_sk(i-1,j-1,k+1)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D3_sk(i-1,j+1,k)==1)&(D3_sk(i+1,j-1,k)==1)
%                     temp(i,j,k)=1;
%                 end
%                 if (D3_sk(i,j+1,k)==1)&(D3_sk(i,j-1,k)==1)

```



```

end
if (sk_3D(i,j,k-1)==1)&(D2_sk(i,j,k+1)==1)
temp2(i,j,k)=1;
end
if (sk_3D(i,j+1,k-1)==1)&(D2_sk(i,j-1,k+1)==1)
temp2(i,j,k)=1;
end
if (sk_3D(i+1,j-1,k-1)==1)&(D2_sk(i-1,j+1,k-1)==1)
temp2(i,j,k)=1;
end
if (sk_3D(i+1,j,k-1)==1)&(D2_sk(i-1,j,k+1)==1)
temp2(i,j,k)=1;
end
if (sk_3D(i+1,j+1,k-1)==1)&(D2_sk(i-1,j-1,k+1)==1)
temp2(i,j,k)=1;
end
end
end
end
sk_3D = sk_3D + temp2;
% repairing discontinuities in direction perpendicular to the slices
temp3 = zeros(s1, s2, s3);
for i=2:(s1-1)
for j=2:(s2-1)
for k=2:(s3-1)
if (sk_3D(i,j,k)==0)
if (D3_sk(i-1,j+1,k-1)==1)&(D3_sk(i+1,j-1,k+1)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i,j+1,k-1)==1)&(D3_sk(i,j-1,k+1)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i+1,j+1,k-1)==1)&(D3_sk(i-1,j-1,k+1)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i-1,j+1,k)==1)&(D3_sk(i+1,j-1,k)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i,j+1,k)==1)&(D3_sk(i,j-1,k)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i+1,j+1,k)==1)&(D3_sk(i-1,j-1,k)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i-1,j+1,k+1)==1)&(D3_sk(i+1,j-1,k-1)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i,j+1,k+1)==1)&(D3_sk(i,j-1,k-1)==1)
temp3(i,j,k)=1;
end
if (sk_3D(i+1,j+1,k+1)==1)&(D3_sk(i-1,j-1,k-1)==1)
temp3(i,j,k)=1;
end
end
end
end
sk_3D = sk_3D + temp3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% cleaning up isolated voxels
isolated = zeros(s1, s2, s3);
for i=2:(s1-1)
for j=2:(s2-1)
for k=2:(s3-1)
if (sk_3D(i,j,k)==1)
counter = sk_3D(i-1,j-1,k+1)+sk_3D(i-1,j,k+1)+sk_3D(i-1,j+1,k+1)+ ...
sk_3D(i,j-1,k+1)+sk_3D(i,j,k+1)+sk_3D(i,j+1,k+1)+ ...
sk_3D(i+1,j-1,k+1)+sk_3D(i+1,j,k+1)+sk_3D(i+1,j+1,k+1)+ ...
sk_3D(i-1,j-1,k)+sk_3D(i-1,j,k)+sk_3D(i-1,j+1,k)+ ...
sk_3D(i,j-1,k)+sk_3D(i,j,k)+sk_3D(i+1,j,k)+ ...
sk_3D(i+1,j-1,k)+sk_3D(i+1,j,k)+sk_3D(i+1,j+1,k)+ ...
sk_3D(i-1,j-1,k-1)+sk_3D(i-1,j,k-1)+sk_3D(i-1,j+1,k-1)+ ...
sk_3D(i,j-1,k-1)+sk_3D(i,j,k-1)+sk_3D(i,j+1,k-1)+ ...
sk_3D(i+1,j-1,k-1)+sk_3D(i+1,j,k-1)+sk_3D(i+1,j+1,k-1);
if (counter == 0)
isolated(i,j,k) = 1;
end
end
end
end
sk_3D = sk_3D - isolated;
% part = sk_3D(21:80,21:80,21:80);

CL = 1

% finding the nodes in the skeleton which are accessible-pore centers
% node = zeros(100,100,150);
% for i=2:99
% for j=2:99
% for k=2:149

```

```

% if (sk_3D(i,j,k)==1)
%   counter = sk_3D(i-1,j-1,k+1)+ sk_3D(i-1,j,k+1)+ sk_3D(i-1,j+1,k+1)+ sk_3D(i,j-1,k+1)+ ...
%   sk_3D(i,j,k+1)+ sk_3D(i+1,j-1,k+1)+ sk_3D(i+1,j,k+1)+ sk_3D(i+1,j+1,k+1)+...
%   sk_3D(i-1,j-1,k)+ sk_3D(i-1,j,k)+ sk_3D(i-1,j+1,k)+ sk_3D(i+1,j-1,k)+ ...
%   sk_3D(i-1,j-1,k-1)+ sk_3D(i,j-1,k)+ sk_3D(i-1,j,k-1)+ sk_3D(i-1,j+1,k-1)+ ...
%   sk_3D(i+1,j,k)+ sk_3D(i,j-1,k-1)+ sk_3D(i+1,j+1,k)+ sk_3D(i,j,k-1)+ ...
%   sk_3D(i+1,j-1,k-1)+ sk_3D(i+1,j,k-1)+ sk_3D(i+1,j+1,k-1);
%   if (counter >= 3)
%     node(i,j,k) = 1;
%   end
% end
% end
% end
% end
% ND = sum(node(:))
% sk_node = sk_3D + node;
%sk_node = sk_node(21:80,21:80,21:80);

% converting mat file into VolView readable .vox format
%filename = 'sk_3D.vox';
%sk_3D = permute(sk_3D,[2,3,1]);
%sk_3D = flipdim(sk_3D,3);
%sk_3D = flipdim(sk_3D,1);
%fid = fopen(filename,'wb');
%fwrite(fid,sk_3D,'double');
%fclose(fid);

%filename = 'sk_3D_part.vox';
%sk_3D_part = permute(sk_3D_part,[2,3,1]);
%sk_3D_part = flipdim(sk_3D_part,3);
%sk_3D_part = flipdim(sk_3D_part,1);
%fid = fopen(filename,'wb');
%fwrite(fid,sk_3D_part,'double');
%fclose(fid);

%filename = 'c_sk_3D.vox';
%c_sk_3D = permute(c_sk_3D,[2,3,1]);
%c_sk_3D = flipdim(c_sk_3D,3);
%c_sk_3D = flipdim(c_sk_3D,1);
%fid = fopen(filename,'wb');
%fwrite(fid,c_sk_3D,'double');
%fclose(fid);

%filename = 'cl_c_sk_3D.vox';
%cl_c_sk_3D = permute(cl_c_sk_3D,[2,3,1]);
%cl_c_sk_3D = flipdim(cl_c_sk_3D,3);
%cl_c_sk_3D = flipdim(cl_c_sk_3D,1);
%fid = fopen(filename,'wb');
%fwrite(fid,cl_c_sk_3D,'double');
%fclose(fid);

%filename = 'sk_node.vox';
%sk_node = permute(sk_node,[2,3,1]);
%sk_node = flipdim(sk_node,3);
%sk_node = flipdim(sk_node,1);
%fid = fopen(filename,'wb');
%fwrite(fid,sk_node,'double');
%fclose(fid);

% VV=1

```