A NETWORK MODEL FOR PERFUSION CHROMATOGRAPHY

by

KAI-CHEE LOH

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

CHEMICAL ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1995

Signature of Author.....
Department of Chemical Engineering
May 11,1995

Certified by.....
Daniel I.C. Wang
Thesis Supervisor, Department of Chemical Engineering

Accepted by.....
Robert E. Cohen
Chairman, Committee for Graduate Students

A Network Model for Perfusion Chromatography

Kai-Chee Loh

Submitted to the Department of Chemical Engineering
on May 11, 1995 in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

The current intense focus in perfusion chromatography for protein purification is due to its increased speed at which separation of proteins occurs. In order to fully exploit the potential of perfusive packings, an understanding of the correlation between the pore size distribution (PSD) and pore interconnectivity (PI) of the particles and the performance of a column packed with such particles is needed.

A network model was used to represent the porous structure existing in a column packed with perfusive stationary phase. By matching model simulations with experimental mercury intrusion data, characteristics of the pore size distribution (PSD) of perfusive particles were calculated quantitatively. It was found that the distribution is bimodal, consisting of macropores (pore diameters of the order of 1000's Å) interconnected by micropores (pore diameters of the order of 100's Å). Two different types of perfusive packings (POROS I and II) were compared in terms of capacity and throughput capabilities with respect to their PSD. The sensitivity and validity of the model was corroborated by simulating intrusion experiments on blends of the perfusive packings. The network model was used to elucidate pertinent parameters using perfusive packings such as surface area accessibility of various proteins and size exclusion chromatography of polystyrene molecules.

Permeability of the perfusive medium was studied by incorporating a transport methodology into the network model. The predictions of the network model were validated with calculations from existing macroscopic models. Simulations to probe the effects of changes in the PSD on intraparticle convection were performed. Based on these, an empirical correlation, which relates intraparticle convection to the PSD and PI, was developed. Intraparticle convection was found to be a strong function of PI and the macropore size distribution while it is weakly dependent on the mean sizes of the micropores and inversely proportional to the spread of the micropore size distribution at constant bed and particle porosities. Using the empirical correlation, it is possible to predict *a priori*, the intraparticle convective velocity. An optimization study reveals that two independent variables can be varied for optimization purposes - the mean macropore size and the standard deviation of the macropore size distribution.

Thesis advisor: Professor Daniel IC Wang
Title: Chevron Professor of Chemical Engineering

A Network Model for Perfusion Chromatography

Kai-Chee Loh
Thesis Advisor: Daniel IC Wang
May 11, 1995

Technical Summary

The current intense focus in perfusion chromatography for protein purification is due to its increased speed at which separation of proteins occurs. The crux of perfusion chromatography lies in the packing media, tradenamed POROS®, the porous structure of which has never been fully characterized. In order to fully exploit the potential of POROS® packings in liquid chromatography, an understanding of the correlation between the pore size distribution (PSD) and pore interconnectivity (PI) of the particles and the performance of a column packed with such particles is definitely needed. This thesis presents the development and experimental validation of a network model which can be used to systematically characterize the PSD and PI of perfusive media. By incorporating a transport methodology into the network model, effects of the PSD and PI on intraparticle convection can be studied. This provides the framework necessary for optimizing the porous structure of the packing with respect to adsorption capacity and throughput.

A lattice network model was established for a representative section of the packed column and mercury intrusion was simulated using the model. Quantitative values of the input parameters (PSD and PI) for the model were calculated from matching the simulated mercury intrusion data to experimentally obtained mercury porosimetry curves. The results indicate that the PSD of POROS® packings comprises of a group of macropores (1000's Å range) and a group of micropores (100's Å range). The model established provides a useful tool for obtaining the PSD of porous materials when other methods like mercury porosimetry involve erroneous data interpretation and SEM and densitometry measurements are too laborius. Using the model, the pore number distribution was discretized on a per particle basis which provides a simple semi-quantitative comparison between different perfusive materials with regards to adsorption capability and throughput. Mercury intrusion was simulated for blends of POROS I and II. The excellent agreement between simulation and experimental data not only corroborated with the network model established, but it also confirms the PSD parameters obtained. More importantly, the results indicate that the network model can be used as a quality control/assurance tool.

By simulating the adsorption capacities for proteins of varying sizes on POROS I and II, the network model was used to study the accessible porous surface area available for adsorption. Excellent agreement was obtained between the experimental data and model predictions. The importance of pore interconnectivity on the accessibility of the porous structure was demonstrated by comparing the accessible surface area for the network model against a parallel pore model whereby the pores are not connected to each other.

Furthermore, it was shown that there is no competitive advantage of POROS II over POROS I, in terms of adsorption capacity, when the solute molecule exceeds a critical size. Additional validation of the model was achieved by simulating size exclusion chromatography (SEC). The agreement between simulation and experimental data indicates that the model can be used to simulate SEC.

The dependence of column efficiency, HETP, on the pore structure of the packings was demonstrated by residence time distribution studies on columns packed with POROS I and II and blends of the pure materials. Experimental results reveal that the particle porosity alone cannot characterize the packing and that the PSD should be used. For perfusive packings with a lower intraparticle velocity, the HETP studies demonstrate that a higher superficial velocity is necessary to trigger intraparticle convection.

Permeability of the perfusive medium was studied by incorporating a transport methodology into the network model. The split ratios of flow rates and superficial velocities in the particle to those in the column were elucidated for packed beds of POROS I, II and blends of the pure packings. The predictions were validated with calculations using existing macroscopic models. The excellent agreement between the split ratios predicted by the network model and the macroscopic models corroborated with the transport methodology used. Simulations to probe the effects of changes in parameters describing the PSD on the split ratio were performed using the validated transport methodology in the network model. Parameters of the PSD and PI were varied and the effects of such variations on the magnitude of intraparticle convection were studied. Based on the study, an empirical correlation which relates intraparticle convection to the microscopic parameters of the PSD and PI was developed. Intraparticle convection was found to be a strong function of PI and the parameters of the macropore size distribution while it is weakly dependent on the mean sizes of the micropores and inversely proportional to the spread of the micropore size distribution at constant bed and particle porosities. Furthermore, the simulations indicate that the Blake-Kozeny equation may not be appropriate for quantifying intraparticle permeability accurately. Experimental data reveal that intraparticle permeability may vary as pore diameter raised to an exponent of 2.58-2.73 as opposed to an exponent of 2 as indicated by the Blake-Kozeny equation. Using the empirical correlation developed and knowing the parameters which characterize the PSD of the packing, it is possible to predict *a priori*, the intraparticle convective velocity.

An optimization study was attempted using the empirical correlation developed. Tradeoffs between surface area and intraparticle convection were illustrated with the conclusion that intraparticle convection has to be sacrificed for surface area and vice-versa. Based on the results of the network model, it was found that there is a flexibility of having two independent variables which can be varied for optimization purposes - the mean macropore size and the standard deviation of the macropore size distribution.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all who have contributed to the completion of this thesis and all who have made my stay at M.I.T. a very enjoyable and memorable experience.

First and foremost, my deepest appreciation to my thesis advisor, Daniel Wang. This research would not have taken shape if not for Dr Wang's brilliant guidance and foresight. I have learnt much from his discipline, demand for excellence and sincere dedication to work. My gratitude is also extended to the members of my thesis committee, Charles Cooney, William Deen, Gregory Stephanopoulos and Noubar Afeyan for their insightful and constructive criticisms. I have benefitted much from their invaluable advice. Special thanks to Noubar Afeyan for giving me much encouragement and support throughout this project.

I am grateful to PerSeptive Biosystems, Inc. for the supply of materials needed in this research. Inasmuch as the materials were important, the contributions from various staff members of PerSeptive Biosystems are deeply appreciated. In particular, I would like to thank Krishna Kalghatgi and Mark McCoy for the many intellectual discussions on protein chromatography. My appreciation also goes to Dennis Morin, who was always ready with the packing media and who helped with the packing of my HPLC columns; Duncan Whitney and his research group for volunteering much of their time and advice to media characterization and not forgetting Rohin Mhatre for helping me with size exclusion chromatography. I hope our paths will cross again.

I also wish to thank Howard Stone (Harvard University) and Sidney Redner (Boston University) for their invaluable suggestions on network modelling. They are indeed exemplaries of an unselfish attitude for the sake of education and research.

My appreciation goes especially to my labmates and officemates: Christine Moore, Eric Scharin, Margaret Speed and Bavy (Stefan Winkler) who

have given meaning to graduate school for me. I will remember you guys for many, many years to come. I would like to especially acknowledge Christine for her encouragement, advice and intellectual contributions. While I miss all the discussions (fuitful or futile) and the friendship we've shared, I hope that we will remain in constant contact in future years. My other colleagues from Building 16 and Building 20 past and present (who are too numerous to mention) have also been a source of encouragement, support and friendship. I would also like to acknowlege the friendship and listening ears of Alexandros Koulouris and David Chang. They were always available at times when I needed someone to talk to. Our encounters may be brief but the memories will last forever.

Thanks are also extended to the office staff at BPEC headquarters: Audrey Childs, Sonia foster and Lynne Lenker for their help in all admininstrative chores through these years. They have been so friendly, helpful and efficient that it's hard to imagine how BPEC can "survive" without them.

Fellowship support from the National University of Singapore is appreciated. I am very grateful to the National University for giving me the opportunity to pursue a PhD overseas and the chance to experience life in the US. Financial support from the National Science Foundation under the Engineering Research Center Initiative to the Biotechnology Process Engineering Center under the cooperative agreement EEC-88-03014 is also acknowledged.

Last, but not least, I would like to thank my wife, Mui-Kim, and her family. This thesis would not be possible without them and is therefore dedicated to them. Mui-Kim's family has been a great source of emotional support and I thank them for having confidence in my abilities. I cannot thank Mui-Kim enough for her unwavering love, sacrifice and patience. She has provided me with an entirely new meaning to love and family. No matter how tough life can be, I can always count on her for support and encouragement. In fact, mere words cannot express how much I am indebted to her and how much I appreciate what she has done for me. Thank you so much, honi

Table of Contents

List of Figures

List of Tables

Chapter 1 -

Introduction

Liquid chromatography has progressed greatly during the last two decades and to date, high performance liquid chromatography (HPLC) has become the unsurpassed technique for isolating and purifying proteins and peptides. With the advent of biotechnology, liquid chromatography now plays an important role not only on the analytical scale, but also on the preparative and production scales. The current focus in large-scale liquid chromatography is on increasing the speed at which separation of protein occurs. Activities in which speed of analysis is important include maximizing production and optimizing isolation on the preparative scale; and confirming biosynthetic fidelity, structural conformation and the absence of host contaminants, collecting data for feedback control in production, examining interactions with other species, determining production shelf life and following peptide metabolism on the analytical scale.

Despite the limited success of using direct scaleup of conventional HPLC packings, there is an increasing need to seek new approaches. Conventional

HPLC packed columns suffers from a number of limitations. The most important of these is the restriction to low flow rates, which stems from the intrinsically low intraparticle diffusive mass transport. In order to enhance the kinetics of the separation process, Afeyan and his coworkers (Afeyan *et.al.*, 1990) have developed perfusion chromatography which takes advantage of packing materials with large through-pores (pore diameters of the order of 1000's Å). Under suitable conditions, a sufficient level of pressure-driven convective flow exists in these pores so that intraparticle mass transfer is augmented by convective transport. To circumvent the problem of low adsorption capacity due to low surface area, the large through-pores in the particles are intercalated with smaller diffusive pores (pore diameters of the order of 100's Å) resulting in a bimodal pore size distribution (PSD) for these perfusive particles.

Theoretical analyses of the effects of intraparticle convection within these large-pore stationary phases have been developed with particular emphasis on the enhancement of column efficiency at high flow rates (Carta and Rodriguez, 1993; Frey *et.al.*, 1993; Rodriguez *et.al.*, 1993, 1991; Liapis and McCoy, 1992; Afeyan *et.al.*, 1990). These analyses have involved macroscopic modelling of the columns which lump pore structure characteristics (PSD) into the empirical tortuosity factor in the constitutive equations. Moreover, all the models developed have assumed that the perfusive packings consist of a bimodal PSD where convection takes place in the macropores (through-pores) and diffusion in the micropores (diffusive pores). Although the characterization of pore structures of HPLC packing materials is important (Tanaka *et.al.*, 1991), especially when perfusion chromatography owes its success to the significantly different PSD of the packings compared to traditional diffusive stationary phases, to date, there has not been any data in the literature that provides

quantification of perfusive particles with respect to the PSD (the relative proportions of macropores to micropores, the mean macropore and micropore diameter). This type of characterization can provide a framework to correlate the PSD of the particles to the performance of a column packed with such particles. This relationship has to be understood in order to fully exploit the potential of large-pore packings in large-scale liquid chromatography.

Porous packing materials have customarily been characterized in terms of PSD by methods such as nitrogen adsorption, mercury intrusion, size exclusion chromatography (SEC) and transmission electron microscopy (TEM). In using the first three techniques, the traditional method of analyzing the resultant experimental data relies on the use of a "parallel-pore" model. In this model, the porous structure of the particles is taken to be a bundle of parallel cylindrical tubes of varying diameters but of the same length. The flaws in this approach are well documented (Dullien, 1992) and are largely due to the fact that the parallel pore model neglects the existence of the interconnections between different pores in the porous media. In the case of using TEM to characterize porous materials, Tanaka and his coworkers (Tanaka *et.al.*, 1991) have found that TEM, in combination with densitometric measurements can be used to visualize pore structures with the possibility of a three-dimensional representation. Their method, however, is very laborious and it is not straightforward in providing a correlation between the characteristics of the porous structure and the performance of an HPLC column packed with the particles.

Discrete pore network models, which incorporate specific geometric properties of the pore space and takes into account the interconnections

between the pores have been developed to provide a more realistic representation of the pore space within porous media (Imdakm and Sahimi, 1991; Portsmouth and Gladden, 1991; Tsakiroglou and Payatakes, 1991; Androutsopoulos and Mann, 1978). In such network models, the porous structure is represented by a lattice network (two-dimensional or three-dimensional) of nodes interconnected by bonds representing the pores. The network is generated using a Monte Carlo simulation approach whereby cylindrical pores are randomly assigned to the bonds of the network. In some models known as pore-throat models (Lane *et.al.*, 1986; Lapidus *et.al.*, 1985), the porous structure is represented in the lattice network by large voids defined as spheres and narrow throats interconnecting the spherical voids defined as cylinders. Such models, however, have been criticized as being too restrictive besides being not very realistic. Furthermore, such network geometry makes it difficult to translate the results to diffusion and reaction simulations.

The diameters of the assigned pores are distributed according to a PSD which has to be inferred by simulating mercury intrusion on the models and comparing the simulated results to the experimental data (Portsmouth and Gladden, 1991; Tsakiroglou and Payatakes, 1991; Conner *et.al.*, 1983; Mann *et.al.*, 1986). Variations of different network models have been used with great success for interpreting mercury porosimetry data and more realistically inferring the PSD of the porous media. In addition, these models permit the direct study of transport phenomena within the pore space (Hollewand and Gladden, 1992; Sahimi *et.al.*, 1991; Rege and Fogler, 1988, 1987) and therefore facilitates the direct correlation of the effects of pore structure characteristics and transport behaviour of the particles.

The goal of this thesis centers on establishing a discrete pore network model to represent a perfusion chromatography column. The network model will serve as a framework with which to gain insight into the relationship between PSD of the stationary phases and intraparticle convection: the cornerstone of perfusion chromatography. Subsequently, guidelines to improve on the microstructure of the media can be established using the network model.

Specifically, the goals of this thesis are:

(1) establish a network model to represent the packed media using a Monte Carlo simulation approach;

(2) obtain characterization data on the PSD of perfusive packings by simulating mercury porosimetry on the model;

(3) validate model with mercury porosimetry on blends of perfusive materials;

(4) use of model to elucidate experimental data pertaining to accessible porous surface area, and size exclusion chromatography;

(5) establish transport methodology to study intraparticle convection;

(6) validate transport methodology with plate height experiments, and existing theoretical approaches;

(7) establish guidelines for the effects of pore structure on column efficiency.

This thesis is organized into seven chapters. In addition to demonstrating the importance to characterize the porous structure of wide-pore materials as used in HPLC, a thorough review of the relevant literature, which is presented in Chapter 2. This will provide the necessary information on the network model formulation and their applications to porous media. The experiments required to characterize and validate the network model are detailed in Chapter 3. In the

experiments, two different perfusive materials will be used. In Chapter 4, formulation of the network model to represent the HPLC column will be presented. The simulation methods used in conjunction with the model will also be described. Chapter 5 reports on the results obtained from simulating mercury porosimetry experiments to obtain the input parameters for the network model. The two perfusive materials used in the experiments will be used to validate the model. Sensitivity analyses and validation of the model using mercury porosimetry experiments on blends of the perfusive materials will be presented and discussed. The utility of the model to predict experimental data based on surface area accessibility of various proteins and size exclusion chromatography of polystyrenes will also be demonstrated. Chapter 5 will also present results on using the network model to elucidate data on intraparticle convection in the perfusive materials. The chapter concludes with simulations designed to probe the effects of changes in parameters describing the PSD on column efficiency. The final two chapters summarize conclusions reached as a result of this research and directions for further research.

Chapter 2 -

Literature Review

2.1 Evolution of Stationary Phases used in Liquid Chromatography

Liquid chromatography has become an indispensable tool for the purification and identification of high-value products from biotechnological processes. In fact, most of what is known today about biochemical processes would never have been discovered without chromatographic separations. A great deal of the success with liquid chromatogaphy can be attributed to the minimization of bandspreading through the design and production of new packing materials. Since the introduction of liquid chromatography in the early 1950s, packings used have been in a continuous process of improvement and refinement. It has been identified that this advancement in design of the support matrices has occurred distinctly in four stages (Behrens, 1992; Afeyan *et. al.*, 1991), namely soft gels, silica supports, polymeric resins and kinetically enhanced resins. These will be concisely reviewed to provide an indication of where the designs of future packings are heading.

2.1.1 Soft Gels

Polysaccharides such as cellulose, cross-linked dextrans and agarose dominated the chromatography of proteins for more than twenty years since their introduction as chromatographic supports in the late 1950s. Cellulose was introduced as an ion-exchange matrix in 1954 by Sober and Peterson (1954), paving the way for the technique to be used in protein separations. Cellulose occurs naturally and is therefore inexpensive. However, it was discovered in later years that the use of cellulose as a chromatographic matrix has some severe disadvantages. To begin, the formation of crystalline-amorphous regions leads to heterogeneity in the matrix. The amorphous regions swell to greater extents and allow deeper penetration of large molecules into the matrix. In addition, the lack of crosslinking in the matrix leads to a non-rigid material which compresses easily under pressure. This limits its use in high pressure or large-scale chromatography systems. Finally, cellulose is susceptible to degradation by many microorganisms which produce cellulases. In using cellulose, it is imperative that the growth of such bacteria or fungi be prevented.

In a search for more rigid matrices, Porath and his co-workers (Porath and Lindner, 1961; Porath and Flodin, 1959) developed a bead form of crosslinked polydextran which has large enough pores to allow proteins and other macromolecules access to the interior of the matrix. By modifying the crosslinked polydextran with suitable functional groups, they had demonstrated the improvements of this soft gel for ion-exchange chromatography. At about the same time, polyacrylamide and agarose were introduced for use in size exclusion chromatography (Hjerten, 1964; Hjerten and Mosbach, 1962). Although the extensive crosslinking of the materials has rendered increased

mechanical stability to the extent that some agarose-based columns will tolerate pressure differentials of 10-20 atmospheres. This was achieved, however, at the expense of decreased porosity and increased hydrophobicity of the matrix.

2.1.2 Silica Supports

Giddings (1964) made a major contribution to the evolution from soft gel materials in the use of inorganic microparticles for use in chromatographic separations by developing a theoretical basis for the technique of liquid chromatography. This led to a better understanding of the fundamental aspects of chromatographic methods. One of the significant contribution of his theory was the realization that separation performance can be improved by using smaller particles at high pressures. This prompted the development of faster separations by Haller (1965). Fine particles of controlled-pore glass were used to achieve the separation of macromolecules in less than 10 minutes. In early 1970s, macroporous, microparticulate silica matrices as small as a few microns were introduced to increase both the resolution and speed of chromatographic columns for the separation of proteins (Unger *et. al.,* 1973; Tsui and Robinson, 1976; Chang *et. al.,* 1976; Ohlson *et. al.,* 1978).

Although these materials were able to withstand high pressures in high performance liquid chromatography (HPLC), biomolecules tended to bind irreversibly to these matrices. The breakthrough came in 1974 when Schecter (1974) treated silica gel with Carbowax to decrease the binding ability of free silanol groups. Since then, various methods of surface coating to protect against non-specific adsorption in silica-based materials have been developed (Regnier and Noel, 1976; Engelhardt and Mathes, 1977). However, there is always a tradeoff with using these silica supports. The main disadvantage with

these materials stems from their limited stability above pH 9. This is a particular liability when cleaning and sterilizing the columns require much higher pH's. Nonetheless, these silica-based media have had an enormous impact on protein chemistry and biotechnology, and continue to be widely used today.

2.1.3 Polymeric Resins

The third phase of development in packing design began during the late 1970s with the introduction of polymeric matrices for HPLC. Through advances in suspension polymerization technology, macroporous polymers such as polymethacrylate, polyacrylamide and poly(styrene-divinylbenzene) can be synthesized with both superior mechanical and chemical stabilities (Coupek *et. al.,* 1973; Hashimoto *et. al.,* 1978; Majors, 1993, Hashimoto, 1991a,b). Moreover, the degree of non-specific binding associated with polymeric materials is considerably less than that for commonly used inorganic materials such as silica. As in the case of silica-based packings, these polymeric materials can be synthesized in particle sizes as small as a few microns and with pore diameters varying up to 1000Å or more (Frechet, 1993; Barrett, 1975; Cheng *et. al.,* 1992a,b). Currently, a variety of polymeric packing materials is available for almost any chromatographic applications. The hydrophobic nature of these materials renders them ideal for use in size exclusion chromatography in non-aqueous solvents such as tetrahydrofuran, or in reversed-phase chromatography with an appropriate choice of solvent.

For other applications, chemical modification may be used to increase the polarity of the resins for use in the retentive modes of chromatography: ion-exchange chromatography, immobilized metal affinity chromatography and biospecific affinity chromatography (Unger, 1990).

2.1.4 Kinetically Enhanced Resins

As the biopharmaceutical industry matures, there is an increasing demand for large-scale protein purification using liquid chromatography. While the three stages of packing development have focussed on improving the mechanical strength, the chemical stability and increasing the surface area available for protein adsorption, little attention has been paid to overcoming the problem of stagnant mobile phase mass transfer in the porous chromatographic packing materials. This motivation prompts the fourth stage of modern packing development.

The principal limitation of the conventional supports described before is their inability to operate at high flow rates due to diffusional problems, while maintaining a reasonable pressure drop, loading capacity and resolution. The technique of HPLC using these packing materials is therefore limited mainly to analytical uses.

In liquid chromatography, solute transport through the column occurs mainly by convective transport in the mobile phase as it flows between the particles in the column and diffusive transport through stagnant liquid pools surrounding the column packing. It has long been recognized that the bottleneck lies in the mass transport to and within the inner surfaces of the packing particles where most of the binding capacity lies (Scopes,1988; Snyder and Kirkland, 1979; Giddings, 1965).

When the time required to diffuse in and out of the pores is greater than the time taken to flow past the particle, the technique suffers from increased

bandspreading, loss of peak resolution and loss of binding capacity. These phenomena would be most serious in the case of macromolecules such as proteins. The key to progress in the liquid chromatographic world is henceforth to suggest a way to rapidly and efficiently expose the mobile phase and the associated solute molecules to the entire surface area within the particle matrix.

In an effort to overcome the mass transport problem, Unger (1974) reduced the particle diameter so as to reduce the effective length of the diffusive path. This, consequently, reduced the time for the solutes to diffuse in and out of the particle. With smaller porous particles, however, the major pitfall is the dramatic increase in the pressure drop through the column at a given flow rate. Moreover, it is quite difficult to pack a uniform column with smaller particles.

On the other extreme, Kalghatgi and Horvath (1988) and Hashimoto (1991) approached the problem by the omission of the pores. In their chromatographic media, all binding occurs on the outer surface and the resistance to diffusive transport is eliminated. In their work, analytical separations under 60 seconds with 2-3μm non-porous particles at mobile phase velocities of 3-5 mm/s were experienced. However, the use of non-porous particles is not a practical solution in preparative and larger scale chromatograhies. Apart from rapid transport, high loading capacity must be maintained for high throughput separations. Non-porous convective packings suffer from having low surface areas and therefore low binding capacities. When beads are made very small to increase the external surface area, the inherent problem of increased pressure drop reappears.

The use of microporous or ultrafiltration membranes which have been surface-derivatized to give selective binding may seem to solve the problem of low surface area at the expense of pressure drop. The surface area of these separation membranes is completely accessible by convective transport. Unfortunately, these have low binding capacity per unit volume and throughput problems associated with membrane fouling and concentration polarization effects limit their applications in high resolution separations.

Recent studies have shown that there is a better solution to the problem of stagnant mobile phase mass transfer. Afeyan and his co-workers (1990) have developed a new chromatographic packing material tradenamed POROS® and the system of chromatography using POROS® media termed "perfusion chromatography". POROS® particles consist of a network of large pores (6000-8000Å in diameters), called throughpores, which allow liquid flow through the sorbent. These are interconnected by micropores (500-1500Å in diameters) through which diffusional transport takes place. In perfusion chromatography, transport into the particles occurs by a combination of convection and diffusion. The advantage of this perfusion process is that intraparticle flow convectively transports solutes to the moiety of the stationary phase in the interior of the particle. Studies have shown that the convective transport of proteins into 10- and 20-μm particles at high mobile phase velocity was found to be an order of magnitude more rapid than diffusive transport (Afeyan, 1990). They have shown that proteins can be separated in less than a minute at mobile phase velocities 10-30 times higher than those used in HPLC by using POROS®-based supports (Afeyan *et. al.,* 1990a,b; 1991). Since perfusion chromatography is based on fluid mechanics, the rate of mass transfer is enhanced without altering

the separation mechanisms. As such, any mode of chromatographic separation has been carried out with perfusable packings.

Fulton and his co-workers (1991), taking advantage of the characteristics of rapid protein separation in POROS® media, developed an affinity chromatographic assay for IgG using protein A and protein G ligands bound to the perfusive particle. The procedure allowed complete assay cycles to be typically performed in 0.5-2 minutes. Moreover, dilute samples such as cell culture media as well as concentrated samples can be readily analyzed on the same system without any dilution or concentration. On the track of high speed assay, Evans *et. al.* (1994) has also developed a flow injection fluorescence immunoassay for the anti-epileptic drug, phenotoin, using a perfusive support immobilized with protein A. It was found that the total time for the assay including the incubation time, was just under 3.5 minutes and the lifetime of each of the POROS®-A immunoreactor was well in excess of 300 runs compared with the traditional 15 minutes assay using controlled pore glass-protein-A columns. In yet another application, Lehman and his team of researchers (1993) demonstrated the use of strong cation-exchange and reversed-phase perfusion chromatographic media to purify tick anticoagulant peptide from a recombinant yeast culture.

It is interesting to note that apart from POROS®-based supports, other large-pore media (Lloyd and Warner, 1990; Hjerten *et. al.*, 1989; Hegedus, 1988) have also been introduced as means to improve on the kinetics of HPLC separations. Similar to perfusive media, these large pore materials have the common important feature of intraparticle convection-augmented mass transfer.

From this review on the evolution of the stationary phases used in liquid chromatography, it seems clear that the direction is towards media which not only has mechanical strength, chemical stabillity, high capacity but also high porosity of convective pores to construct a more efficient column with high resolution. An important area which is lacking, however, is the characterization of the large-pore materials with respect to the pore size distribution existing in the media. Afterall, the success of perfusion chromatography over conventional HPLC is supported by the distinctly different porous structure in the perfusive media. The next section provides a brief review of some of the common techniques used for the characterization of porous media in terms of their pore size distribution.

2.2 Pore Size Distribution Characterization Techniques

A knowledge of the pore size distribution of bulk packings largely enables the chromatographer to understand and interpret the retention mechanisms of chromatographic processes through an appropriate correlation. This provides useful guidelines for choosing a suitable packing for a given separation problem and enables the chromatographer to differentiate between various types of packings on a rational rather than purely empirical basis. Customarily, the pore size distribution of porous packing materials has been characterized by various experimental methods. These are capillary condensation, mercury porosimetry, size exclusion chromatography, and electron microscopy. The fundamentals and a review of these classical methods will be presented.

2.2.1 Capillary Condensation

Capillary condensation has been used to calculate the distribution of pore sizes, mostly using the "bundle of capillary tubes" model. In the "bundle of capillary tubes" model, the porous structure is assumed to be a group of parallel, cylindrical capillary tubes of different diameters and equal lengths. Every capillary has a uniform diameter along its entire length. The fundamental equation used in relating pore size to vapor pressure is Kelvin's equation (Equation (2.1)).

$$d_p(\text{Å}) = \frac{8.28}{\log \frac{P_o}{P}} \qquad \text{...(2.1)}$$

where d_p is the pore diameter

P_o is the vapor pressure, and

P is the equilibrium pressure of the adsorbate, usually nitrogen

Capillary condensation, also termed gas adsorption, employing nitrogen as adsorbate, is the method of choice for determining the distribution of pores in sizes less than 500Å. After outgassing the sample at the appropriate temperature and thermostatting at the temperature of liquid nitrogen (77K) , the adsorption and desorption branches are obtained over the whole range of the isotherm. The analysis of the pore size distribution is based on the isotherm data reported by Brunauer and his co-workers (1967), using the *t*-plot of a non-porous reference material of the same surface chemical composition (Unger, 1979; Mikhail and Robens, 1983; Shull, 1948). The calculation gives the distribution of the specific pore volume or specific surface area of the microporous material as a function of the diameter, as given by Equation (2.1).

Since capillary condensation sets on at an already adsorbed multilayer by exceeding a relative pressure saturation $P/P_o > 0.4$, the Kelvin pore diameter must be corrected for the thickness of this multilayer. Wheeler (1946) was apparently the first to point out the necessity of accounting for the thickness of this multilayer.

Since then, numerous calculational methods have been developed to assess the pore size distribution (Barett *et. al.,* 1951; Crenston and Inkley, 1957; Dollimore and Hed, 1964). They differ in the choice of the adsorption or the desorption branch of the isotherm, and in the assumption of a variable or constant thickness of the adsorbed multilayer. The various authors usually compared, as a check, the total surface area obtained by their respective techniques with the value obtained by the BET method (Brunauer *et. al.,* 1938).

Shull (1948) and Barett *et. al.* (1951) used the desorption isotherms and obtained good agreement with the BET surface areas in most cases. Crenston and Inkley (1957), however, performed a more comprehensive study of this question by comparing the surface areas calculated from both the adsorption and the desorption branches of the isotherms with the BET areas. His comparison showed, on the average, better agreement in the case of adsorption isotherms. The surface areas calculated by Bernardini *et. al.* (1967) were strikingly different from the BET areas. Voight and Tomlinson (1955) used the adsorption branch for porous Vycor glass, where the pores are mostly of the so-called ink-bottle type. They obtained good agreement with the BET surface area. They also demonstrated a considerable variation in the calculated surface area depending on what adsorbate was used.

It is clear from these experimental evidence that whether the adsorption or desorption branch of the mercury intrusion curve is to be preferred for pore size distribution calculations depends on the type of porous material . It is noted also that agreement with BET surface areas does not prove that the pore size distribution is accurate. In terms of the accuracy of the measurements in the experiment, it was found that the accuracy decreases with increasing relative pressure, so that in practice it is only possible to use the method, on a relative basis, for a pore diameter range of 20 to 500Å (Unger, 1990). In addition, the model of pore structure used in the method, the "bundle of capillary tubes" model, is a highly idealized model. Consequently, any increased precision of the calculations can be expected to result in very little real improvement in the accuracy of the pore size distribution obtained.

2.2.2 Mercury Porosimetry

The first mercury high pressure porosimeter constructed by Ritter and Drake (1945) 50 years ago made it possible to determine the pore size distribution by measuring the volume of mercury intruded into the pores as a function of the pressure employed. The pore volume distribution is then calculated on the basis of the Washburn's equation:

$$d_p = -\frac{4\gamma\cos\theta}{P} \qquad \text{...(2.2)}$$

where d_p is the pore diameter

γ is the surface tension of mercury (480 dynes/cm)

θ is the contact angle of mercury with the solid (145^{o})

P is the intrusion pressure

Interpretation of the intrusion data into pore size distribution is primarily performed in terms of the "bundle of capillary tubes" model which is inaccurate for a porous network of highly interconnected pores. Apart from assuming that the cylindrical pores are of uniform diameter throughout the length of the pore, interconnections between the pores are also not taken into account. As a result, the intrusion volume of mercury at each intrusion pressure is assigned to the diameter of the entry pores (throats) through which the pores are accessible. The intrusion curves normally do not give any information on the size of the pores that are accessible through the throats in an interconnected porous structure.

The intrusion process involves the sequential invasion of the pore structure by penetration through the throats. However, it can generally be assumed that the majority of the internal void space is found in the pores; the volume of the throats is negligible (Conner and Lane, 1984). A throat is measured when mercury breaks through a throat and into a pore at a specified applied pressure. This event is recorded by measuring the volume intruded into the pore. Penetration of a throat into an already filled pore will therefore not significantly affect the intruded volume and the throat is effectively not measured.

The analysis is not even that simple. Not all throats of a penetrable size (hence measured) are accessible at any given time. Some of the throats are "shadowed" are therefore not accessible until a pore to which the throat is connected has been filled. Throats, even large throats at the interior of the particle, are not measured until the adjoining pores have been invaded. After mercury has intruded a specific pore through a throat, new throats are made

accessible. If any of these throats are larger than the throat that was intruded, they are also penetrated with mercury. These will, however, be measured at a pressure corresponding to the smaller throat of access. The "shadowing" phenomenon results in measuring the throats smaller than they actually are.

In addition, there is also a "non-linkage" phenomenon concerning pores which are linked to a filled pore at the time of their penetration and were too small to be penetrated previously . These are either not recorded or they are measured as larger than they actually are.

This can be envisioned by considering the intrusion process into a two-dimensional network as shown in Figure 2.1. The structure is depicted to consist of pores (shown by circles) interconnected by the throats (labelled I, S and L). The first throats penetrated should be the large throats, labelled S. However, these sit in the shadow of the smaller ones labelled I and L which have access to the bulk mercury. Penetration will occur only when the pressure corresponding to throat I is reached. The mercury front will, at this time pass through throat S, but they will be measured at too high a pressure (shadowing). At this point, the entire void structure is filled. Therefore, intrusion of throats L at a higher pressure will not significantly affect the intruded volume, and hence they are not measured (non-linkage). It is also partly because of this interconnections between pores of differing sizes that hysteresis is often observed in mercury porosimetry (Wardlaw and McKellar, 1981; Dullien, 1981; Bell *et. al.*, 1981). In fact, distortion in the pore size distribution calculated based

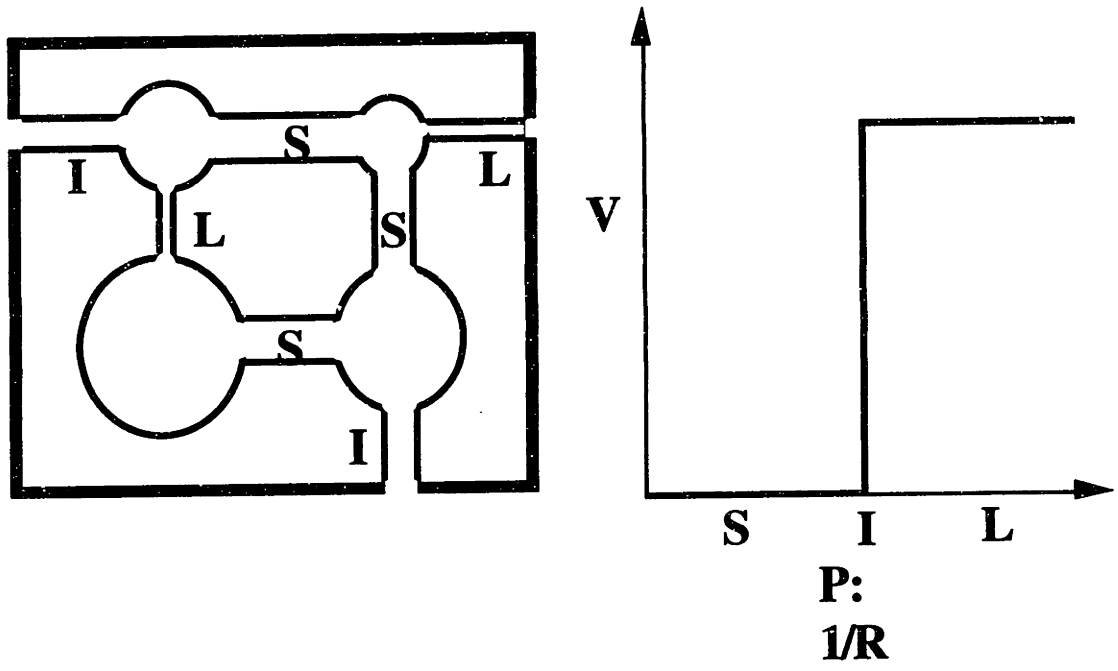


Figure 2.1 Mercury intrusion into an interconnected two-dimensional network. P represents the intrusion pressure and V is the cumulative intrusion volume. Shadowing effect results in measuring pores at smaller diameters, while the linkage problem results in measuring pores at larger diameters.

on the "bundle of capillaries" model have been demonstrated experimentally by many researchers (Dullien, 1975; Dhawan, 1972; Batra, 1973).

Notwithstanding these experimental verifications of the inadequacy of mercury porosimetry for even characterization of the distribution of pore and volume by the appropriate pore dimensions, owing to its simplicity, the method is still widely used. However, the pore size distribution calculated from mercury intrusion curves should be used with discretion.

2.2.3 Size Exclusion Chromatography

Size exclusion chromatography of standard polymers on porous packings has been developed as an alternative tool for assessing pore size distribution (Halasz and Martin, 1978; Knox and Scott, 1984; Niclov *et. al.,* 1980; Werner and Halasz, 1980; Vilenchik, *et. al.,* 1993). The possibility of the determination of the pore size distribution of porous materials ensues from the equilibrium distribution of rigid molecules of the solute under conditions of the exclusion of molecules worked out by Giddings and his co-workers (1968). In 1978, Halasz and Martin (1978) defined the boundary conditions necessary for the technique and demonstrated the possibility of its utilization with swelling sorbents based on polymers (Kuga, 1981). For the practical solution of the problem of pore size distribution in macroporous styrene-divinylbenzene copolymers, this method was applied by Freeman and Poinescu (1977) who also introduced the term "inversion gel permeation chromatography". The pore size of the material and the size of the solvated polymer solute are approximated by appropriate models (Kreveld and van der Hoed, 1973; Volkenstein, 1963; Schultz and Baumann, 1968; Giddings *et. al.,* 1968); these then enable the chromatographer to correlate the elution volume of a polymer

solute with the pore size and to calculate the pore size distribution. The size exclusion chromatography method is quick and simple to use and the measurements are usually carried out under the usual conditions of application of the porous media, ie. chromatography. Moreover, high pressures (mercury porosimetry) and low temperatures (capillary condensation) are unnecessary. Interpretation of the data, however, relies on the fact that the elution volume of the polymer represents the porous volume of all pores of the same size as that of the polymer used in the experiment which is obviously erroneous in the case of an interconnected pore network. This limitation is very similar to the shadowing effect seen in mercury porosimetry.

In addition to the inaccurate interpretation of the data which stems from the pore model selected, two major drawbacks with regards to resolution have been found. Calculations have shown (Yau *et. al.,* 1979) that even in the case of a packing with uniform pores of equal size, the calibration curve, log molecular size versus elution volume, spans almost 1.5 decades of the molecular weight. Moreover, the whole spectrum of pore sizes, from 10Å to10 μm, is matched by 10 to 15 data points, corresponding to the number of narrow molecular weight standards employed. As a consequence, the distribution calculated from the elution volumes on this basis does not provide a high resolution.

2.2.4 Electron Microscopy

To a limited extent, electron microscopic methods have been used for the direct determination of the properties of packing materials. Grimaud and his co-workers (1978) used scanning electron microscopy (SEM) in comparative ultrastructural studies of three series of gels: Ultrogel, Sephader G and Bio-Gel P. In SEM, photomicrographs of the outer surfaces of the packings are obtained

for visualization of the porous structure. To obtain the pore size distribution, a great many of such photomicrographs have to be taken for a representative quantification. Moreover, the definition of a pore is, in many sense very subjective. Since SEM is usually performed for the external surfaces, care has to be taken to correlate the external porous structure to the overall pore size distribution of the particle. Nonetheless, SEM has been used in various occasions for quantitative comparisons between porous packings (Unger and Gimpel, 1979; Colin and Guichon, 1976; Tracz *et. al.*, 1984; 1986;).

Another electron microscopic method for pore size distribution characterization is transmission electron microscopy (TEM). In this method, sections of the porous material is visualized under TEM. Tracz and Barna (1986) have reported the preparation of these ultra-thin samples by ion milling and Tanaka *et. al.* (1988, 1991a, b) have demonstrated the use of an ultramicrotome for preparing the samples. Sections of ca. 80nm have been used to observe the internal structure of various packing media. By coupling the TEM technique to densitometry measurements of the thin samples, Tanaka and co-workers (1991a, b) have indicated that the three-dimensional internal structures of the adsorbents can be reconstructed by serially cutting the particles into many sections. Their method, however, is very laborious and the resolution of the technique is questionable; the resolution depends on how thin the sections of the samples can be obtained. Currently, only 80nm sections have been used and therefore only pores larger than a few hundred nanometers (a few thousand angstroms) can be observed.

Although all of the above methods used to characterize the pore size distribution of porous packings suffer from individual disadvantages, the data

obtained can be useful if discretion and care are exercised in the interpretation of the data. More often than not, these data serve as initial inputs for modeling techniques in an effort to more accurately define the porous structure. In what follows, different modeling approaches will be reviewed.

2.3 Modelling Approaches

Many processes of engineeering interest involve the physicochemical interaction of a fluid with the solid surface of a disordered porous medium. Examples include filtration operations, wood and coal combustion and gasification, heterogeneous catalytic reactions, gel permeation and liquid chromatography. All of these processes are characterized by a continuous alteration of the pore structure either by chemical reaction or physical interaction (pore plugging or particle adsorption and deposition) of the particles and the solid surface of the pores. In some cases, such as protein separation by chromatography and gel permeation chromatography, the closeness of the sizes of the particles and the pores gives rise to the so-called hindered or restricted transport in porous media (Deen, 1989; Sahimi and Jue, 1989), which is also an important technical problem.

Basically, there are two major approaches which have been used to model these processes. They are the continuum and discrete approaches (Sahimi et al, 1990).

2.3.1 Continuum Models

In a continuum model, the porous material is treated as a continuum within which temperature and species concentrations are defined as continuous

functions of time and position. These pointwise functions are usually governed by the appropriate mass and energy balances. When the continuum model is used to describe materials of complex and irregular geometry, differential equations for momentum, energy and mass and the associated boundary conditions at the fluid-solid interface are first formulated. However, as the interfaces of disordered media are highly irregular, practical and technical means are not available for storing such boundary value problems. Even in the unlikely event that the detailed morphology of the porous medium is known, the boundary within which the equations of transport and reaction have to be solved would be so tortuous that the problem will be rendered mathematically intractable. More often, the solution of the problem (assuming that it can be obtained!) could contain much more information than would be useful practically.

In a continuum model, macroscopic properties such as effective transport coefficients and reaction rates are defined as averages of the corresponding microscopic quantities. However, in many situations, the validity of averaging is not satisfied. When applied to disordered media, derivation of the macroscopic transport coefficient is often difficult because of the complex geometry of the porous medium. In addition, these models are also not well suited for describing drastic changes in the connectivity of the medium and underlying phenomena like pore plugging and fragmentation (Rege and Fogler, 1988). Continuum models assume complete connectivity and concentrate only on the geometry of the pore space.

The difficulty with using these models is the presence of many empirical parameters; the estimation of which is not always possible due to a lack of

sufficient experimental data. Moreover, some of these parameters are not even well-defined or have no physical interpretation (Imdakm and Sahimi, 1991).

Nevertheless, continuum models are still widely employed because of their convenience and familiarity to the engineer. The simplest model used is the so-called "grain model" in which the solid phase is characterized by small spherical particles or grains of a specified radius. Szekely and his co-workers (1975) used this model to study coal-char gasification whereby the basic constitutive equations were formulated such that gas concentration and solid conversion are functions of time and position of the grain within the pellets. However, the representation of the solid structure by a uniform local length scale (grain radius) is clearly seen as an over-simplification. In a subsequent paper, Szekely and Propster (1975) extended the calculations to include a grain size distribution to overcome this particular difficulty and concomitantly, the model provided more versatility.

In a variation of the model, the porous particles were modeled by cylindrical capillaries to represent the pores (Simons and Finson, 1979; Hashimoto and Silveston, 1973). These models made use of the Carmen-Kozeny equation to evaluate permeability factors and then introduced a tortuosity factor to correlate with the experimental data. In addition, they have not attempted to describe the geometry of capillary intersection and overlap and have treated the evolution of pore volume and surface area by *ad hoc* relationships. Feng et al. (1974), and Wang and Smith (1983) extended the capillary model by including multicomponent diffusion and allowing a random orientation of the capillary axes. However, the material balance equations at the intersection of different capillaries were violated because the interaction of

fluxes at those points was neglected. Therefore, although the models are relatively easy to use and sufficiently accurate, it holds rigorously only for networks of infinitely long, straight and non-overlapping capillaries.

The use of continuum, macroscopic models, however, has met with numerous success for analysing intraparticle convection in porous media. In the simplest application, Afeyan *et. al.* (1990) formulated an analysis based on the Carmen-Kozeny equation to derive the intraparticle velocity in the particle as a function of the diameter of the throughpores in perfusive media. Park and Stephanopoulous (1992) used a modification of the Blake-Kozeny equation to describe and experimentally demonstrate the existence of intraparticle convection in ceramic particles used for the immobilization of mammalian cells.

Using the macroscopic differential equations for conservation of mass on both the particle level and the packed column level, Rodriguez and his co-workers (Rodriguez *et. al.,* 1991; 1982; 1992a; b; c; Carta *et. al.,* 1992) derived the concept of a convection-augmented effective diffusivity to account for the effects of intraparticle convection on bandspreading characteristics in an HPLC column. Their analyses resulted in the development of a generalized van Deemter's equation accounting for intraparticle convection. Experimental observations reported by Afeyan *et. al.* (1990) reported that the HETP (height equivalent to a theoretical plate), a measure of the column efficiency, tends to a plateau at high superficial velocities in the presence of intraparticle convection corrobarated with the developed generalized van Deemter's equation. Using a similar approach, Frey *et. al.* (1993) also developed a continuum model to examine the effectiveness of large-pore column packings for rapid analytical

chromatography and for the concentration and recovery of a dilute solute in a saturation-regeneration cycle utilizing frontal chromatography.

In addition to analysing plate height characteristics, effects of intraparticle convection on peak resolution in linear chromatography (Rodriquez, 1993) and adsorption chromatography (Carta and Rodriquez, 1993) were studied using macroscopic modeling. All of these continuum models, however, are based on the assumption that the adsorbent particles consist of a bimodal pore size distribution whereby convective throughpores are intercalated with diffusive micropores. Characteristics of the pore morphology and topology are lumped into empirical factors of tortuosity and porosity.

Liapis and co-workers (Liapis and McCoy, 1992; McCoy *et. al.*, 1991; 1993) have also developed a mathematical model to describe the dynamic behavior of single and multicomponent adsorption in columns packed with perfusive media.

Related work in the field of reaction engineering was discussed by Lu *et. al.* (1992), Nir and Rismen (1977), Prince *et. al.* (1991), Rodriquez and Ferreira (1988) and Rodriquez *et. al.* (1982). They have shown that the effectiveness factor of a large-pore catalyst for isothermal, irreversible first-order reactions can be enhanced by intraparticle convection in the intermediate region of the Thiele modulus. In the biochemical engineering area, the idea was also demonstrated by Stephanopoulos and Tsiveriotis (1989) in relation to porous pellets formed by cell aggregation as in the fermentation of mycelial organisms. However, as in all continuum macroscopic models, it is not straightforward how the results emanating from such analyses relate specifically to the characteristics of the

pore size distribution existing in the porous media. For an understanding of such correlation, discrete models are usually employed.

2.3.2 Discrete Models

In a discrete model, the medium is treated as a a collection of discrete elements such as voids, crystallites, and so on. The fluid and solid species concentrations are now defined for each element and it is again governed by the necessary mass and energy balances. The main difference between the continuum model and the discrete model is therefore whether or not macroscopic properties (temperature, concentrations) are treated as continuous functions of position or as a collection of properties attached to the discrete elements constituting the material. Usually, the discrete model is based on a network representation (Fatt, 1956). Although the original network representation of the pore space is rather old; it was only recently that systematic and rigorous procedures were developed (Mohanty *et. al.,* 1982). In principle, the disordered porous medium is mapped onto an equivalent random network of interconnected pore bodies and throats using statistical methods. Once the mapping is completed, transport and reaction processes in porous media and other appropriate systems can be studied in a more realistic manner. In addition, by using statistical approaches, it is possible to study in great details fluid-solid operations in a porous medium where the effects of topology are naturally taken into account and *a priori* knowledge of effective transport coefficients is not required. In fact, the effective transport coefficient can be obtained from such models (Imdakm and Sahimi, 1986a, b; 1991).

The continnuum model does not provide any insight about the dependence of the transport coefficients on the morphology of the pore space

(like its pore size distribution and interconnectivity). In the case of a discrete model, the importance of pore space interconnectivity (PI), which naturally incorporates the existence of narrow necks, dead ends, the existence of threshold value for transport through the pore space, the relationships between open and total porosity (that is, the fraction of total porosity which is accessible to diffusion) can be accounted for (Sahimi and Jue, 1989; Reyes and Jensen, 1985). However, computations involved in discrete models are always computer-intensive and computer storage problems are often encountered (Hoshen and Kopelman, 1976).

Nonetheless, discrete models are extensively used to study transport processes in porous media. Some examples on the use of network modeling in chemical engineering processes will be reviewed next. The emphasis here is on demonstrating the widespread applicability and success of discrete models to predict and elucidate experimental data in a number of different processes. Here, the use of network modeling for simulating mercury porosimetry will also be presented. This provides the background for the development of the model used in this research.

In many applications of discrete modeling, effective transport coefficients in disordered porous media were determined (Sahimi and Jue, 1989; Akanni *et. al.,* 1987; Abbasi *et. al.,* 1983; Benzoni and Chang, 1984; Berman *et. al.,* 1986; Brandt, 1975; Burganos and Sotirchos, 1987; Nicholson and Petropoulos, 1971; 1975; Sahimi and Imdakm, 1988; Reyes and Jensen, 1985; Nakano and Evans, 1985; Evans *et. al.,* 1980; Petropoulos *et. al.,* 1991; Petrou *et. al.,* 1990).Reyes and Jensen (1987, 1988) used a network model to study and describe the formation damage resulting from particle entrapment in porous media by

straining or size exclusion. Using the model, they were able to account for the reduction in permeability of the media which could not be accounted for using the continuum model (Sharma and Yortsos , 1987). Similar network models were used by Heiba *et. al.,* (1982) together with percolation theory to estimate permeabilities and capillary pressures in two-phase flow through porous media while Sahimi *et. al.,* (1983, 1986a, b) studied hydrodynamic dispersion in one- and two-phase flow through random networks. Imdakm and Sahimi (1987) also developed a network model to study the transport of large particles in flow through porous media, which proves to be an improvement over the model proposed by Sharma and Yortsos (1987), which uses an effective medium approximation in conjunction with a network model of the pore space.

In applications to reaction engineering, Reyes and Jensen (1985) used a network model of pore topology, a Bethe lattice, to simulate pore space related properties in their work on catalytic combustion. The model provided a clear understanding of the role of narrow necks, tortuous paths and dead ended pores on the effective diffusivity which could not be obtained from the traditional tortuosity factor description. From the model, they also found that the intra-catalyst particle diffusivity measured under reactive conditions is different from that under inert conditions. This is similar to the findings of Otani and Smith (1966) who had reported that the effective diffusivity under reactive conditions was about 20 % of that measured in the absence of any reaction. In contrast, Ryan *et. al.,* (1980) have argued that the effective diffusivity should be independent of the chemical reaction rate.

Other uses of network modeling in the field of reaction engineering takes into account fundamental changes in the structure of the pore space due to

chemical reaction, particle deposition and chemical dissolution of the media (Mohanty *et. al.,* 1982; Shah and Ottino, 1987; Kerstein and Bug,1986; Sandmann and Zygourakis, 1986; Zygourakis and Sandmann, 1988; Sahimi and Tsotsis, 1988; Yortsos and Sharma, 1986; Yu and Sotirchos, 1987; Daccord, 1987; Gladden, 1991; Hollewand and Gladden, 1992). In applying discrete modeling to biotechnology processes, Applegate (1991) developed a network bond model to represent the porous structure existing in macroporous ceramic matrix bioreactors for mammalian cell culture. The model developed was used to describe the interaction between non-uniform cell distribution and oxygen transport in a mechanistic manner. The effects of convective and diffusive transport of oxygen in the ceramic matrix, and changes in the pore structure due to cell immobilization on entrapped cell concentrations, viability and reactor operation were investigated using the network model.

Finally, it was noted that Mojaradi and Sahimi (1988) also used a network model to study diffusion-controlled reactions in a disordered porous medium. Their analysis was carried out for a non-uniform distribution of reactants in the medium (that is, those in which a complete mixing of the reactants has not taken place). For such situations, almost all of the available methods, such as effective medium approximation or the classsical equations of chemical kinetics, fail to describe properly the reaction behavior of the system.

Androutsopoulos and Mann (1979), Mann *et .al.,* (1986a), and Mann and Golshan (1981) had used a two-dimensional network pore model to interpret mercury porosimetry experiments in porous solids. Their models consisted of cylindrical pore segments assembled into square grids and thus introduced the important element of pore interconnectivity. Although these models are

simplified, they described appositely the crucial role of pore connectivity on hysteresis and entrapment of mercury in porosimetry experiments, as well as on the proper interpretation of the pore size distribution. In the use of these models, the researchers have attempted to fit the network prediction of the intrusion curve to the experimental data by varying the parameters of the pore size distribution used in the simulation. However, the two-dimension model has a number of inherent disadvantages; the most important of which is the lack of a third dimension. As such, three-dimension network models of mercury porosimetry were proposed by Wardlaw and McKellar (1981) and Cox (1991). Connectivity effects have been studied by changing the geometry of the network from simple cubic through face-centred cubic to body-centred cubic, or by blocking pores in the network to alter the topology of the lattice.

While the models referred to above are bond networks where the junctions defined by the intersection of pores were given no volume, refinement of these models by introducing a distinction between bonds and sites within the void structure has been used (Wall and Brown, 1981; Lane *et. al.,* 1986; Lapidus *et. al.,* 1985; Tsakiroglou and Payatakes, 1990; Wardlaw *et. al.,* 1987). These models have commonly been referred to as pore-throat network models. Although these pore-throat models have been used with great success to model porous media, simulate mercury porosimetry and demonstrate the presence of hsyteresis and mercury entrapment, they suffer from a major drawback in the inability to directly translate the model to convective-diffusion modeling. This arises because of the pore shapes used to represent the bonds and sites in the network: a sphere to represent the pore situated at a site and a cylinder to represent the throat situated at a bond. For modeling transport phenomena on network models, the bond percolation network is still primarily used. Apart from

using cubic lattices, modifications of the network models which make use of three-dimensional, spherical random networks have also been demonstrated (Gladden, 1991; Hollewand and Gladden , 1992; Hollewand *et. al.*, 1991; Portsmouth and Gladden, 1991). In one study by Hollewand and Gladden (1992), attempts to elucidate the pore-interconnectivity using network modeling based on "Reverberi" loops (Reverberi *et. al.*, 1966) have been shown.

Even though these discrete models represent idealizations of real porous solids, they illustrate an important aspect of diffusional processes in porous solids - no resort has to be made to tortuosity factors, since tortuous paths are already built into the computer simulated solid. This is an important advantage since tortousity factors cannot be determined by theoretically sound arguments, and in most cases can only be poorly estimated. In the next section, a brief outline of percolation theory is described. Percolation theory is often used in conjunction with developing a discrete model and it therefore seems apt to introduce the basic principles involved.

2.4 Percolation Theory

Percolation processes were formally introduced in the mathematical literature by Broadbent and Hammersley (1957). Percolation theory seeks by statistical means to describe the morphology of randomly disordered media and transport therein. Thus, it provides a natural framework for evaluating the role of pore interconnectivity on the effective transport coefficient (Reyes and Jensen, 1985). Percolation theory has been used in numerous modeling applications including epidemiological models for simulating spread of forest fires and epidemic diseases (Cardy, 1983; MacKay and Jan, 1984; Stauffer, 1985),

critical phenomena such as the polymerization of branched macromolecules leading to gelation, ferromagnetic, and thermal phase transitions (Flory, 1941; Stockmayer, 1943; Stanley, 1971; Domb and Green, 1989), diffusion in disorrded media (deGennes, 1976; Pandey *et. al.,* 1984) and chemical reactions in porous media (Sahimi *et. al.,* 1990). Processes which may be modeled percolatively range from the sub-atomic to the galactic. For an excellent overview of percolation theory and its applications, the interested reader can refer to Stauffer and Aharony (1992) and Sahimi *et. al.* (1990).

Traditionally, percolation theory has been used to describe the behaviour of systems around critical phase transition. In fact, the phase transition in critical phase phenomena was one of the original motivations for percolation research (Yang, 1983). In such phenomena, the phase transition of a material in response to a smoothly varying, continuous change in temperature appears discontinuous. The material is a solid at one temperature and a fluid at another, and there is a specific temperature at which the material cannot be described as either a solid or a fluid. In the case of diffusional studies on disordered media, conduction of fluid is a discontinuous function of the porosity of the disordered media. There will be no flow of fluid through the media at low porosities and at high porosities, fluid convection increases to high velocities. Consequently, there is some intermediate porosity for which a single "sample-spanning" path first appears, and the convective velocity at that porosity is non-zero.

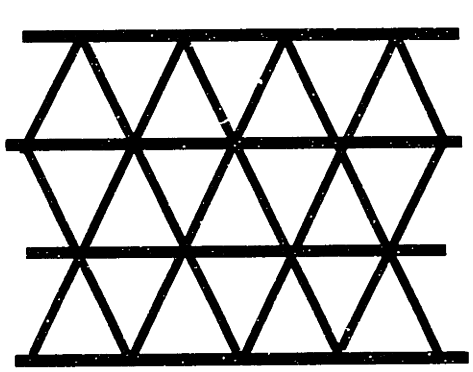
In using percolation theory to model such behaviour, the material or disordered media is represented by a network configuration of sites connected to each other by bonds. The sites are always placed at random and independently from each other on the lattice. In such models, Monte Carlo

simulation techniques are frequently employed. The classical percolation theory therefore centers around two problems - bond percolation problem and site percolation problem. If the volume of the system is assigned primarily to the sites, the model becomes a site percolation problem. Conversely, if the volume of the system is assigned to the bonds, it becomes a bond percolation problem. A typical example of a bond percolation problem is the sol-gel transition (Flory, 1941; Stockmayer, 1943) which involves the formation of bonds between molecules. In the sol state, the solution is a liquid with dispersed molecules of the cross-linked solute species. At a critical number of cross-links, a gel macromolecule spans the system, and the viscosity of the substance increases dramatically and discontinuously. In contrast, consider a lattice of glass spheres between the plates of a capacitor and assuming that the glass spheres are randomly replaced by copper spheres of an equal size. At some characteristic fraction of copper spheres, the capacitor will discharge through a continuous path of copper spheres. In this case, the percolation type is site percolation, since copper spheres occupy individual lattice sites to cause percolation.

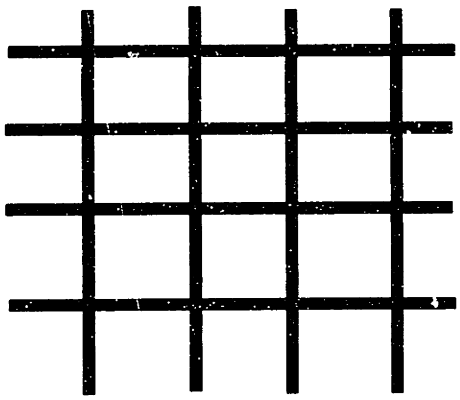
Whereas percolation theory is strictly valid only for systems that are infinitely large, simulations of percolation problems on lattices of infinite size are impossible in a world of limited resources. Fisher (1971) investigated the effect of the finite size of a thermal system on its critical properties near the critical temperature, and was the first to develop a theory of finite-size scaling to extrapolate results obtained using finite lattices to infinite networks. Subsequently, accurate estimates of many percolation quantities have been projected for infinite systems (Reynolds *et. al.*, 1980; Dorrida *et. al.*, 1983; Sahimi *et. al.*, 1983; Stauffer and Zabolitzky, 1986).

In modeling employing percolation theory, different lattice configurations of different geometrical shapes have been used. Some of these configurations are shown in Figure 2.2. and they include triangular, square, honeycomb and Voronoi polyhedron lattices in two dimensions, cubic and other regular polyhedra in three dimensions, and a Bethe lattice in infinite dimensions (Applegate, 1991).

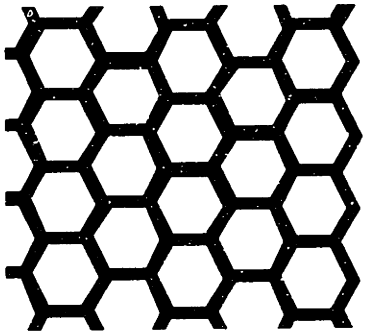
As examples of the use of the different tessallations of space, Sevick *et. al.,* (1988) have used site percolation on a square lattice to study thermal critical phenomena, Pandey *et. al.* (1984) have modeled diffusional processes around the percolation threshold using triangular and cubic lattices and Winterfeld *et. al.* (1981), Kerstein (1983) and Elam *et. al.* (1984) have represented granular media using Voronoi tessalations. Taking advantage of the fact that Bethe lattices are computationally simple, and percolatiom quantities can be obtained analytically, Reyes and Jensen (1986a, b; 1987) have used a Bethe lattice to represent chemical reactions in porous media. Although different researchers have shown preferences for different configurations, many percolation problems are solved using square or cubic lattices due to ease of computation. Winterfeld *et al.* (1981) and Jerauld *et. al.* (1984a, b) have shown that, as long as the average coordination number of a topologically-disordered system is equal to the coordination number of a regular network, the transport properties of the two systems are, for all practical purposes, essentially identical.



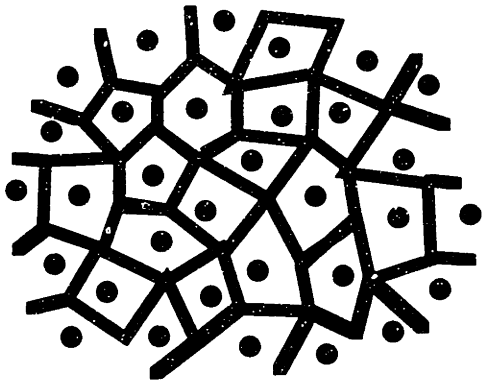
triangular



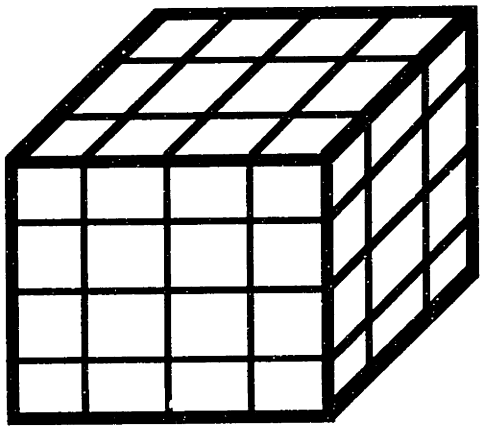
square



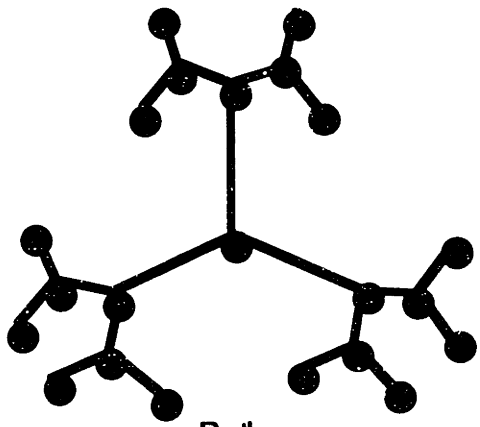
honeycomb



Voronoi



cubic



Bethe

F

igure 2.2 Examples of different lattice configurations used to model percolating behaviour (Applegate, 1991).

This section will be concluded with a definition of the percolation threshold and connectivity which are the cornerstones of percolation theory. Connectivity is defined as the average number of bonds which are connected to a single node, or site. Hence, if a bond (or site) is present in the network with a probability p, the connectivity for that network is pz, where z is the coordination number of the lattice configuration of interest. z is the maximum number of nearest neighbouring sites in site percolation problems or the maximum number of bonds connected to a site in bond percolation problems. Therefore, connectivity values can range from zero to the coordination number of the lattice and is not restricted only to integers.

A cluster refers to a group of connected sites bounded by vacant bonds or open sites. At low values of p, the average cluster size is small, and clusters are isolated from each other. As p increases, the average cluster size grows larger with the coalescence of some clusters. The percolation threshold, p_c is that concentration p at which a cluster of "infinite" size appeared in the infinite lattice. For all $p > p_c$, there will be a cluster extending from one side of the system to the other, whereas for all $p < p_c$, no such infinite cluster exists. The useful and interesting observation about percolation threshold is that it is characteristic of the percolation system. Although measurements of the percolation threshold vary statistically, they are repeatable for a given set of parameters. The parameters include the type of percolation (site or bond), shape of percolating particles (sphere, cylinders, cubes, etc.), lattice type (square, random, etc.) and matrix dimensionality (Stauffer, 1985).

Percolation thresholds for various lattices are tabulated in Table 2.1. Many of these values are estimated through Monte Carlo simulations of finite-

sized lattices. In general, the percolation threshold decreases with increasing coordination number. In addition, bond percolation thresholds are less than site percolation thresholds for the same lattice configuration since bonds connect two sites at a time.

Table 2.1 Selected point percolation thresholds (p_c) for various lattices (Stauffer and Aharony, 1992)

Lattice Type	Coordination Number	p_c (bond)	p_c (site)	Lattice Dimensionality
Chain	2	1.0000	1.0000	1
Triangular	6	0.3473	0.5000	2
Square	4	0.5000	0.5928	2
Honeycomb	3 (6)	0.6527	0.6962	2 (3)
Diamond	4	0.3880	0.4280	3
Simple cubic	6	0.2488	0.3116	3
BCC	8	0.1803	0.2460	3
FCC	8	0.1190	0.1980	3

Chapter 3 -

Materials and Methods

3.1 Materials

Perfusive materials used for the mercury porosimetry, protein adsorption and size exclusion chromatography experiments were POROS R1M (Lot R120-008) and POROS R2M (Lot R220-127) from PerSeptive Biosystems, Inc. (Cambridge, MA, USA). In this thesis, they will be referred to as POROS I and POROS II respectively. As for the permeability studies based on the modified Blake-Kozeny equation, POROS I packings in different nominal diameters were used. These were 10μm (Lot R110-017), 20μm (Lot R120-008) and 50μm (Lot R150-079) packings. POROS particles have been produced through suspension polymerization techniques (Afeyan, *et.al.*, 1990) and classified by mechanical sieving to the desired particle size range. R1M and R2M particles are nominally 20μm in diameter. These are non-derivatized native polystyrene-divinylbenzene materials used for reversed-phase chromatography.

Table 3.1 Proteins used in Protein Adsorption Studies

Protein	Catalog Number	Molecular Weight (kD)	Radius (Å)
Lysozyme	L-6876	14	24
Ovalbumin	A-5503	45	36
Hemoglobin	H-7379	68	41
Transferrin	T-1408	77	43
Fibrinogen	F-9754	340	71

The proteins used for the adsorption capacity experiments were obtained in purified form from Sigma Chemicals (St. Louis, MO, USA). These were chosen such that their molecular sizes range from approximately 50Å to 140Å in diameter. The proteins used, together with their relative molecular weights and effective solute radius as calculated based on diffusivity measurements (Tanford, 1961) are tabulated in Table 3.1.

For the size exclusion chromatography experiments, polystyrene macromolecules were used. Polystyrene was chosen as probes for the experiments because they are available in a variety of sizes with very narrow polydispersivities. The polydispersitivity of a polymer, also known as the heterogeneity index is used as a measure of the width of the molar mass distribution of the polymer. This is defined as the ratio of the weight-average molar mass ($\overline{M}_W$) to the number-average molar mass ($\overline{M}_n$), that is:

$$Polydispersity, I = \frac{\overline{M}_W}{\overline{M}_n} \qquad ...(3.1)$$

The polystyrene standards used were obtained from Supelco (Catalog number 4-5713, Bellefonte, PA, USA). Details of the polystyrene standards are tabulated in Table 3.2. The solvents used in all the experiments, where required, were HPLC grade.

3.2 Mercury Porosimetry

Mercury porosimetry data form the basis for formulating the lattice network model. In this research, mercury porosimetry was performed on the

Table 3.2 Polystyrene Standards used in Size Exclusion Chromatography

Molecular Weight (kD)	Polydispersity, I (as defined in Equation (3.1))
7.5	1.06
25	1.06
47.5	1.06
90	1.04
207	1.06
900	1.10
3000	1.20
6000	1.12

pure materials of POROS I and POROS I I and also on mixtures of POROS I and II in specified proportions.

Mercury intrusion experiments were conducted using a Micromeritics PoreSizer 9320 (Micromeritics, GA, USA) located at E25-342, MIT. During the experiment, the volume of mercury intruded into the sample is measured as a function of intrusion pressure by means of a capacitance detector in the instrument. A powder penetrometer (Part number 920-61714-00) was used since the samples are very fine particles. It is important that a suitable amount of sample is used in each of the experiments. For best resolution, the volume of intruded mercury at the end of the experiment should not exceed 90% or be less than 25% of the maximum measurable intrusion volume of the penetrometer. It was found that a sample weight of about 0.1g would result in a total intrusion volume of about 75% of the maximum measurable intrusion volume of the penetrometer. In all the porosimetry experiments, therefore, about 0.1g of sample was used. In the case of the mixing experiments, POROS I and II were blended in proportions of 25% w/w, 50% w/w and 75% w/w of POROS I in the resulting blends. Four grams of each blend was prepared and for each porosimetry experiment, 0.1g of the blended medium was used.

The sample was weighed out using an analytical balance and poured into the penetrometer with the aid of a small funnel. When pouring the sample into the bulb, a finger was placed over the stem opening in the center of the bulb so that powder did not enter the stem. The penetrometer was then sealed with vacuum grease (Apiezon H). Care was taken to apply a sufficient amount of vacuum grease. Too much grease exposes the sample to an unwanted coating and is likely to cause slippage and misalignment of the mating surfaces of the

penetrometer. On the other hand, too little grease results in an imperfect seal. The mercury intrusion experiment was performed in a succession of two parts - a low pressure analysis followed by a high pressure analysis.

In the low pressure analysis, the sample was first evacuated using a vacuum pump connected to the PoreSizer. During evacuation, care was taken to ensure that the sample did not fluidize. This was achieved by evacuating the sample to a pressure of about 0.5psia, slowly, before switching to a fast evacuation. The evacuation was then continued until the vacuum on the sample stabilized. This vacuum pressure would be less than 50μm mercury. The low pressure analysis was then performed by gradually filling the penetrometer with mercury by manually increasing the mercury pressure from 0.5psia to 30psia. The volume of mercury intruded was recorded as a function of mercury pressure. When the low pressure analysis was completed, the penetrometer was transferred to the high pressure chamber for the high pressure analysis. This was performed in an automatic mode by establishing a tabulated series of pressures ranging from 30psia to 30,000 psia in the control module. The equilibration time needed before a datum point was collected was set at 7-10 seconds as recommended by the vendor. The mercury advancing contact angle used was 140^{o} and the surface tension of mercury was taken to be 485 dynes/cm. For each sample, the mercury porosimetry experiment was performed at least three times to ensure reproducibility. At the end of the intrusion experiment, the particles were inspected under the light microscope for mechanical damage due to the high pressure. It was found that for all the samples tested, there was no visible breakage of the packing materials, even at 30,000 psia.

3.3 Scanning Electron Microscopy and Digital Image Analysis

To determine the type of statistical function to use for the pore size distribution (PSD) in the network model, scanning electron microscopy (SEM) images of the external surfaces of the perfusive materials were obtained. After that, digital image analyses were performed to obtain the frequency data of the particle and macropore sizes of the packing materials.

An SM-510 (Topcon, Avon, CT, USA) scanning electron microscope, located at PerSeptive Biosystems, Inc., was used for the SEM. The sample (POROS I and POROS II) was first applied to double-sided adhesive tape (3M Co., St. Paul, MN, USA) and subsequently affixed to the specimen support. It was necessary to remove loosely stuck particles by hitting the specimen support lightly against a hard surface. The particles were then sputter coated, using the Desk II Sputter (Denton Vacuum Inc., Moorsetown, NJ, USA), also located at PerSeptive Biosystems, Inc., with a layer of about 100Å of gold before being loaded onto the microscope.

SEM measurements were made at 15kV at magnifications of 500x and 10 000x. The lower magnification was used for the particle size distribution analysis while the higher magnification was used to visualize the macropores. Clear, sharp images could not be obtained at higher than 15,000x to visualize the micropores at 15kV. When the voltage was increased to 25kV, the sample was severely burnt by the electron beam. As a result, digital image analysis was only performed for the particle size distribution and the macropore size distribution. At least five frames were taken at each magnification of each

sample for digital image analysis. The procedure for the digital image analysis is described next.

The photomicrographs obtained from SEM were analyzed by digitizing the image. The software, PC Image v1.3 for Windows 3 (Foster Findlay Assoc. Ltd., Newcastle upon Tyne, UK) was used to digitize and analyse the images. The image was captured and transferred to the digitizing board using a Videk MegaPlus Model 1400 camera (Eastman Kodak, Rochester, NY, USA) equipped with a Nikon zoom lens and light aperture(Nikon, Garden City, NY, USA). A TV monitor attached to the system was used to inspect and control the quality of the transferred image.

The image was digitized and each picture point was given one out of 256 possible grey levels. The image digitizing process took about 5 seconds and the information was stored in the computer. The data stored was later translated into binary (logical) information. The grey threshold level settings were found to be crucial for image enhancement. The grey threshold level was changed until the digitized image (as visualized on the TV monitor) best reflected the image obtained from SEM. This, however, could only be determined visually. For the particle size distribution analysis, a grey threshold range of 34-208 pixels was found to be optimum; for the macropore size distribution, the optimum range was from 0 to 45 pixels. In analyzing the macropore size distribution, the effect of surface curvature of the particles have to be taken into account. As such, the region of interest, which represents the area of the image to be digitized, was limited to less than 9μm x 9μm on each frame.

The image was then calibrated based on the magnification of the photomicrograph before analysis was performed. In analyzing the particle size distribution, a circularity cutoff criterion was used to determine the objects (the particles) which have to be counted. The circularity represents the "roundness" of the object and is defined by Equation (3.2):

$$\text{Circularity} = \frac{4\pi.area}{(perimeter)^2} \qquad ...(3.2)$$

where "*area*" is the object area, measured in square pixels, and
"*perimeter*" is the object perimeter, measured in pixels.

A circularity cutoff exceeding 0.80 was used in the image analyses for the particle size distribution. In the case of the macropore size distribution, the circularity criterion is not appropriate and instead, the object area criterion was used. The objects (in this case, the macropores) was counted only when the area of the object exceeds 50 pixels. The area of the counted object was then converted into an equivalent circle of the same area to obtain the macropore diameter.

The image analysis resulted in frequency data as a function of the sizes of the objects (particles or macropores). This frequency data was tested for normality (Gaussian distribution) by making a probability plot of the data set (Taylor, 1990). The data points were arranged in ascending order of their magnitude (particle diameters or macropore diameters), called ranking. The probability plot position, F_i, defined by Equation (3.3) was then calculated.

$$F_i = 100(\frac{i-0.5}{n}) \qquad ...(3.3)$$

where n = total number of data points

i = rank = 1 to n

The particle diameters or macropore diameters were plotted on the ordinate against values of F_i on the probability axis (abscissa) and the normality of the frequency data was judged based on linear regression. The goodness of fit of the probability plot to a straight line indicates the normal distribution of the data obtained.

3.4 Protein Adsorption Capacity

To demonstrate the application of the network model to predict surface accessibility data, experiments of equilibrium adsorption capacity studies were performed on POROS I and II for proteins of various molecular weights and hence molecular sizes.

The static experiments were carried out by mixing 20mg of the particles with different concentrations of protein solutions (0.5, 1.0, 2.0 and 4.0mg/ml) to obtain the adsorption isotherm. Prior to mixing with the protein solutions, the particles were wetted with 40µl of iso-propyl-alcohol (IPA). This was necessary since the native particles are hydrophobic and therefore will not suspend well in the protein solutions. Wetting the particles with IPA, however, does not alter the porous structure in the particles (Kalghatgi, 1994). Two-ml of deionized water was then added to dilute the wetting agent and subsequently the suspension

was centrifuged at 2000rpm for 10 minutes for the particles to settle. As much of the supernatant was removed and the volume of diluted wetting agent which remained with the beads was recorded by taking the difference between the weight of the particles and the weight of the particles with the remaining wetting agent. This volume is required in subsequent calculations of total protein adsorbed on the surfaces of the beads.

Ten-µl of the protein solution was then added to the wetted beads. The suspension of beads and protein was mixed using an orbital shaker overnight. For sampling, 1ml of the suspension was taken and centrifuged at 5000rpm for 6 minutes. The supernatant, representing the unbound protein remaining in solution, was measured using a uv spectrophotometer at 280nm with a reference wavelength of 400nm to account for unsettled particles. In the experiments using hemoglobin, a reference wavelength of 800nm was used because hemoglobin was found to absorb at a wavelength of 400nm. The amount of bound protein, accurate to less than 5% error, was calculated from material balance. For each protein at each concentration, the experiment was performed four times. The experiments which were accepted were those in which the amount of bound protein differed from the average by less than 3%.

In order to check the validity of using the material balance as an accurate measure of the amount of protein bound to the surface, the bound protein was eluted from the particles with 20% acetonitrile (ACN) in 0.1% trifloro-acetic acid (TFA). This validity check was performed only for lysozyme on POROS I since lysozyme eluted cleanly from the particles. For the protein elution, after the suspension of beads and proteins was centrfuged and the supernatant discarded, the particles were washed twice, each with 10ml of water . After the

second wash, the supernatant showed no absorbance reading on the uv spectrophotometer indicating that all the unbound protein had been washed from the surfaces of the particles. Three-ml of the eluent was then added and the suspension mixed for about 30 minutes to ensure that all the proteins had been eluted from the particles. The supernatant was then measured for the amount of bound protein using uv detection after spinning the particles down at 5000rpm for 6 minutes. Table 3.3 summarizes the results obtained from material balance and elution. The error in using the material balance to calculate the amount of bound protein was found to be less than 7%. in all subsequent experiments, the proteins were therefore not eluted, but material balances were used.

The adsorption isotherm obtained experimentally was fitted with the Langmuir-Hinshelwood adsorption kinetics to obtain the maximum static capacity for each protein:

$$q = \frac{q_{\max} S_{eq}}{K_p + S_{eq}} \qquad ...(3.4)$$

where q = protein adsorbed (ml/g)

$q_{\max}$ = static capacity (ml/g)

S_{eq} = equilibrium protein concentration (ml/g)

K_p = affinity constant of protein for particle surface (ml/g)

3.5 Size Exclusion Chromatography

The utility of the network model to elucidate experimental data based on volume accessibility was investigated by size exclusion chromatography (SEC).

Table 3.3 Amount of Lysozyme Adsorbed on POROS I: Comparison between Material Balance and Protein Elution

Protein Concentration (mg/ml)	Material Balance (mg/ml)	Elution (mg/ml)	% Difference
0.5	3.5±0.1	3.7±0.4	7
1.0	7.5±0.2	7.0±0.2	7
2.0	10.0±0.3	10.5±0.3	5

SEC was performed on the Hewlett Packard HP1090 L system (Waldbronn, F.R.G.). The HP1090 L was equipped with a diode array detector, an automatic injector and a "Chem Station" for data acquisition and system control.

Columns (4.6mm diameter x 100mm length) of POROS I and POROS II were obtained prepacked from PerSeptive Biosystems, Inc. (Cambridge, MA, USA). These were flow-packed at 180 bar using a Shandon (Cheshire, UK) packer. The polystyrene standards were prepared in HPLC grade tetrahydrofuran (THF) at 0.1%w/v. The prepared standards were used on the same day because the higher molecular weight polystyrene macromolecules tend to degrade when kept overnight in THF.

Prior to sample injection, the columns were equilibrated with THF. Using the automatic injector, 100µl of sample was applied on the column and the elution profile was recorded using the "Chem Station". A flow rate of 0.40ml/min was used and the eluates were detected at 220nm. For the lower molecular weight standards, since the polydispersivities were narrow, the chromatographic peaks had good symmetry and small bandspreading. As a result, the elution volume was obtained as the midpoints of the eluted peaks. As for the higher molecular weight standards (molecular weights > 900 kD), there was slight assymmetry in the peaks and area integration of the peaks was performed to obtain the elution volume. Pre-column volume was determined by acetone injection without the column and elution volumes were corrected for pre-column plumbing. For all the samples, the elution volumes were determined to an accuracy of less than 1.0% in error.

The solute diameters of the polystyrene standards were calculated from the formula (Van Krefeld and Van den Hoed, 1973):

$$D(Å) = 0.246M^{0.588} \qquad ...(3.5)$$

where M is the relative molecular weight of the polystyrene molecule.

The diameters of the polystyrene standards used in the experiment were calculated from Equation (3.5) and shown in Table 3.4.

The majority of chromatographers now agree that size exclusion chromatography can be explained fully on a purely steric basis that large molecules can only partially permeate the pore volume of the stationary phase (Yau, *et.al.*, 1979). Based on this notion, small molecules (such as those of the eluent) fully permeate the pores of the support material and are eluted in the void volume, V_m, (defined as the sum of the interstitial volume, V_o, and the intraparticle volume, V_i); large molecules which cannot enter any of the pores are totally excluded and are eluted in the interstitial volume, V_o; molecules of intermediate size are eluted between V_o and V_m. Hence, the exclusion coefficient, K, of the polystyrene macromolecule was calculated as follows:

Volume of column $= V_c$

Total volume of pores in column $= \varepsilon_t V_c$

Volume of interstitial pores, $V_o = \varepsilon_c V_c$

Volume of intraparticle pores, $V_i = (\varepsilon_t - \varepsilon_c)V_c$

Exclusion Coefficient,
$$K = \frac{V_e - V_o}{V_i} \qquad ...(3.6)$$

where V_e = elution volume of polystyrene solute

Table 3.4 Properties of Polystyrene Standards

Molecular Weight (kD)	Diameter (Å)
7.5	46
25	94
47.5	138
90	202
207	328
900	780
3000	1584
6000	2380

ε_t = total porosity in column (interstitial + intraparticle pores)

ε_c = column porosity

The total porosity in the column was determined by an acetone injection. From mercury porosimetry, the porosity of the particle was found from:

$$\varepsilon_p = 1 - \left(\frac{1}{1 + V_{int}\rho}\right) \qquad (3.7)$$

where V_{int} = specific volume of mercury intruded into beads (ml/g)

ρ = skeletal density of bead particles (1.05 ml/g)

The column porosity was then calculated from:

$$\varepsilon_t = (1 - \varepsilon_c)\varepsilon_p + \varepsilon_c \qquad ...(3.8)$$

3.6 Laser Scattering Technique for Particle Size Distribution

In using the modified Blake-Kozeny equation to study intraparticle convection, perfusive packings of different sizes were used. The nominal diameters of these particles were obtained experimentally using the Horiba LA-900 (Horiba Instruments Inc., Ann Arbor, MI, USA) Laser Scattering Particle Size Distribution Analyzer. The instrument, equipped with a He-Ne laser tube at 633nm, uses the principle of the Mie (Kerker, 1969) scattering theory for measuring the size and distribution of minute particles suspended in a liquid.

A suspension of the sample (0.04mg/ml) was prepared in about 250ml of acetone. Acetone served as a dispersant for the solid particles. It is important

that the dispersant selected properly "wets" the sample powder. The dispersant must also be one that does not cause coagulation of, dissolving of, swelling of or chemical reaction with the perfusive samples. Acetone was found to be a suitable dispersant for the reversed-phase particles. A blank was first measured using just acetone in the sample holder. Two-hundred and fifty-ml of the suspension was then emptied into the sample holder of the instrument and was kept in the dispersed state by agitating and circulating the suspension at levels of agitation and circulation of 4 "units".

A laser beam emitted from the laser and the tungsten lamp light source radiated the particles suspended in the dispersant. After the laser beam had been dispersed and scattered by the particles in the flow cell, the intensity of the scattered light, which was focussed on a photo-cell detector, was converted into electrical signals. These were then used to calculate the size distribution of the particles. The particle size distribution was obtained by deconvoluting the Fraunhofer diffraction of the scattered light. This was achieved using the software on the Horiba LA-900. However, the sampling technique of the instrument is currently inadequate and hence only the mean particle size of the sample can be obtained accurately. At the moment, the vendor is looking into ways to correct the sampling method so as to obtain the standard deviation of the size distribution accurately.

3.7 Plate Height Analysis

In order to study the effects of intraparticle convection on column efficiency , plate height experiments were performed. This is equivalent to the measurement of the residence time distribution (RTD) of an inert tracer in the

column. Packed columns (4.6mm x 100mm) were obtained prepacked from PerSeptive Biosystems, Inc. (Cambridge, MA, USA). Columns of POROS I, POROS II and blended mixtures of the two packings in proportions of 25% w/w, 50% w/w and 75% w/w of POROS I in the resulting blends were used for the experiments.

The RTD of lysozyme at non-retained conditions was performed on each column using the BioCad (PerSeptive Biosystems, Inc., Cambridge, MA, USA) which was equipped with an autoloader, a uv detector and a data acquisition software - BioCad Workstation on the personal computer. For the experiments, the instrument was configured in the single column mode. The tubings used in the plumbing had an inner diameter of 0.010 inch as recommended by the vendor. All solvents used were filtered through 0.22μm filter and degassed before being used. The mobile phase used was 0.1% w/v TFA in 70% v/v ACN/water for non-retained conditions. The test solute, lysozyme, was dissolved in the mobile phase at a concentration of 1.0mg/ml. Prior to sample injection, the system was purged and the columns equilibrated with 5 column volumes of the mobile phase. The lysozyme solution was filtered through 0.22μm filter and 1.0μl of the sample was injected onto the equilibrated column. The flow rate used varied from 100 cm/h to 4000 cm/h to obtain the van Deemter HETP data and the experiments were performed in triplicates.

The resulting chromatogram was detected and recorded at 215nm. The area integration software on the BioCad Workstation was used to calculate the amount of bandspreading according to the following equations:

The number of transfer units, $N = 5.54(\frac{V_e}{W})^2$,

and reduced HETP, $h = \frac{L}{Nd_p}$

where V_e = elution volume of the solute,

W = peak width at half peak height,

L = length of the column,

d_p = diameter of the packing beads

Chapter 4 -

Model Formulation and Simulation Methods

4.1 Method of Approach

This chapter details the steps and considerations taken in developing the lattice representation of the perfusive media in an HPLC column. The simulation methods used for elucidating experimental results in support of the network model will also be described. Figure 4.1 shows an outline of the overall method of approach. Central to this research is the generation of the network model which is a Monte Carlo lattice representation. Inputs required to establish the model include the type of statistical function required to represent the pore size distribution (determined from SEM analysis), the form of the composite PSD (determined by the porous structure existing in the media), type of random number generator, lattice size and pore length. After the network has been generated, mercury porosimetry is simulated to obtain the cumulative intrusion curve. By comparing the simulated and experimental intrusion curves, the

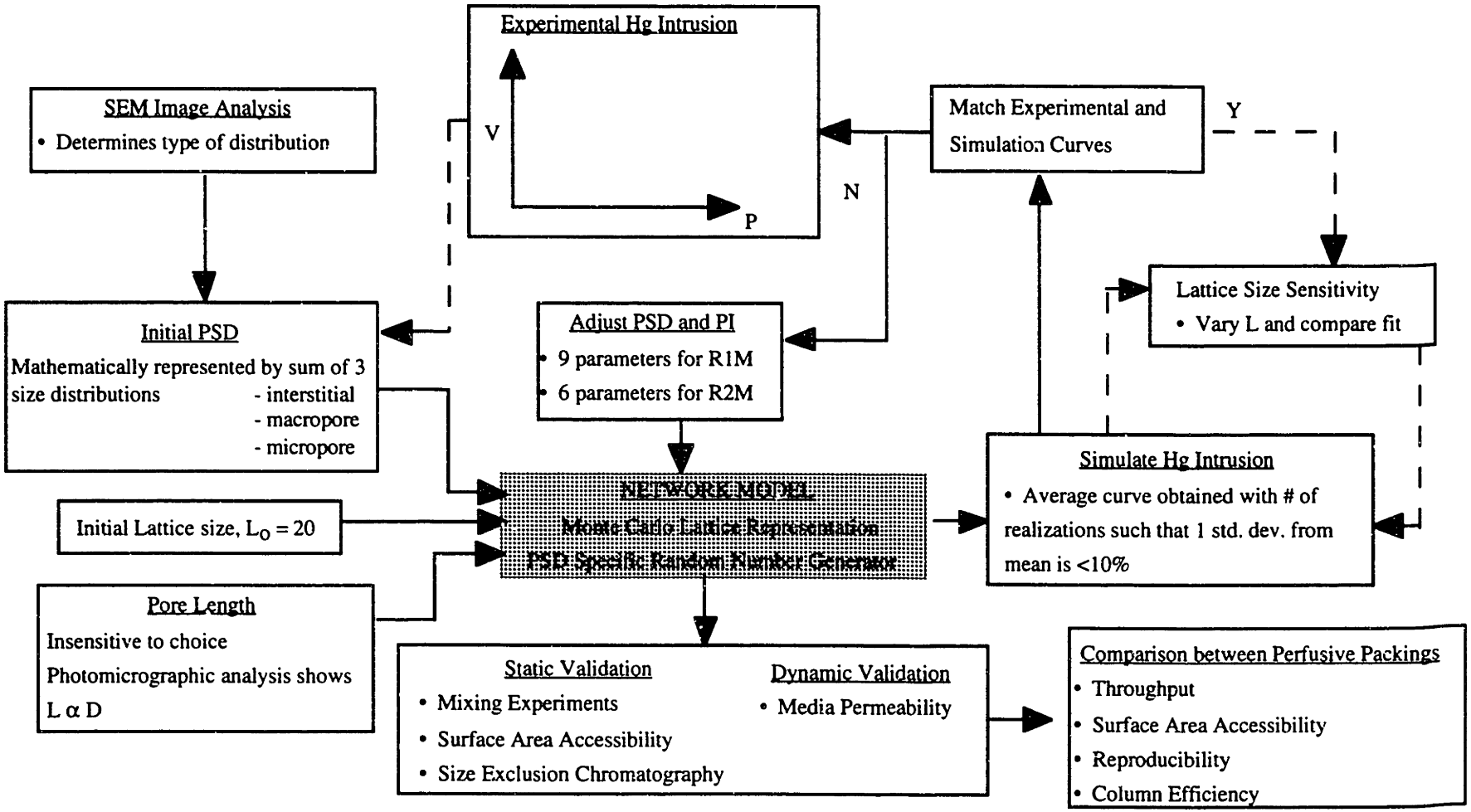


Figure 4.1 Outline of Overall Method of Approach

parameters of the composite PSD of the media are adjusted until a good match is obtained. The parameters are first obtained for POROS I followed by POROS II. Sensitivity analyses are subsequently carried out to ascertain the uniqueness of the PSD's obtained and to study the effects of lattice size. Validation of the network model against experimental results is achieved by simulating mercury intrusion on blends of POROS I and II, surface area accessibility of different proteins, size exclusion chromatography of polystyrene macromolecules and flow permeability. These corroborations will aid in the comparison between different perfusive packings with regards to reproducibility, surface area accessibility, throughput and column efficiency.

4.2 Model Formulation

The greatest challenge in pore structure modelling is to incorporate the elements of randomness and chaos implicit in porous media in such a way as to retain both structural realism and tractable quantitative treatment. In this research, the packed column is represented by a cubic lattice network of interconnected cylindrical pores. In addition to simplicity, the use of a cubic lattice has the added advantage of allowing the numerical calculations to be performed more efficiently. Cylindrical capillary geometry is used because of tradition and convenience in the calculation of the effective diffusivity and other related transport phenomena properties. The lattice used consists of a regular (X x Y x Z: L x L x L) array of nodes connected to each other by bonds of the network (Figure 4.2). To represent the porous media in the column realistically, the bond size distribution of the network should be similar to the PSD existing in the chromatographic media. The porous structure in the HPLC column is classified into two classes of pores - interstitial pores, representing the pores

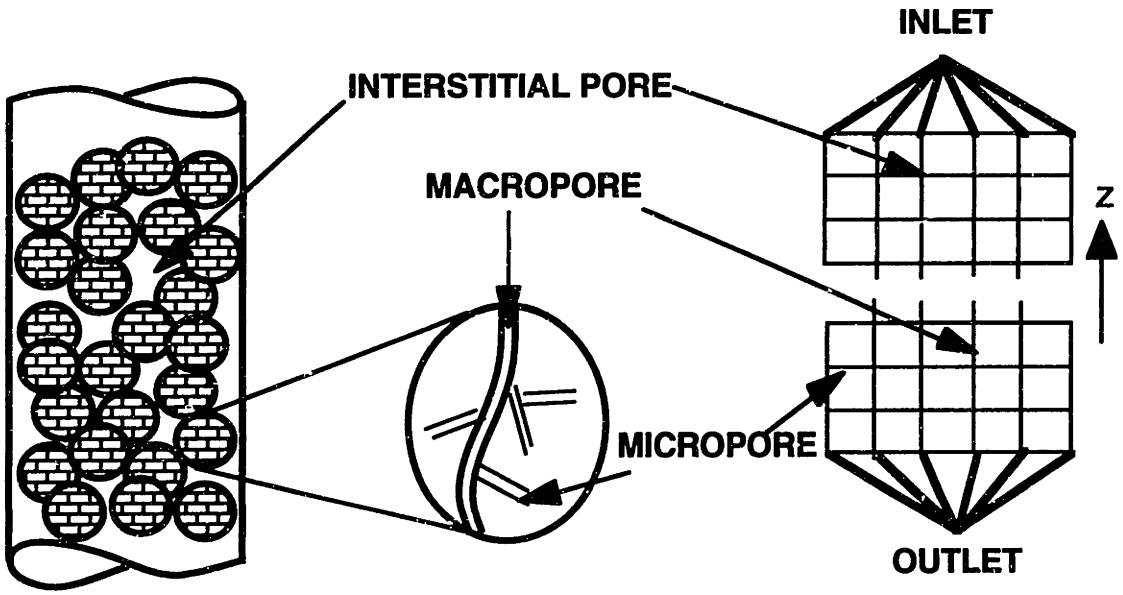


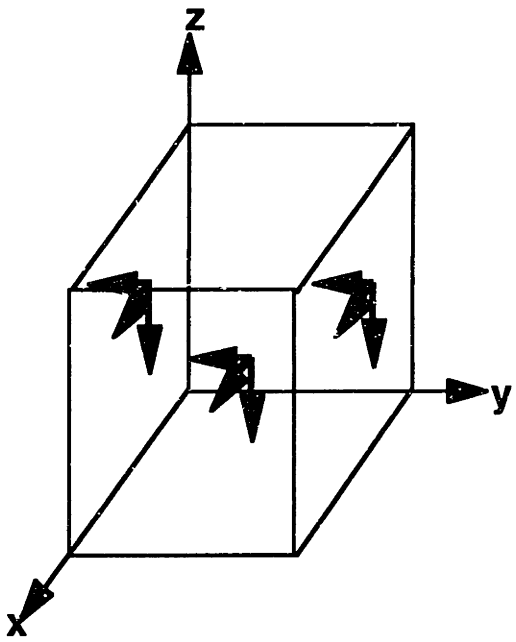
Figure 4.2 Schematic representation of chromatographic media for network model.

between the bead particles and intraparticle pores, representing the pores within the porous beads: the latter being subdivided into two subclasses of pores - macropores and micropores. The "composite" pore size distribution representing the porous structure is mapped topologically onto the lattice shown in Figure 4.2 by randomly assigning cylindrical pores to the bonds in the network. As is the standard practice (Sahimi *et. al.,* 1990), no volume is assigned to the nodes of the network. All the porosity of the system is provided by the pores at the bonds. Mapping of the porous media is accomplished as outlined in Figure 4.3.

Basically, a point in the network (coordinates x,y,z) is selected at random and an orientation of the pore is chosen (orientation pointing to the left, to the front or downward from the node). In this case, only three orientations need to be considered as the other three orientations at a cubic network node will be accounted for by nodes adjacent to the node in question. This cylindrical pore is then assigned a diameter based on a predetermined statistical distribution function representing the PSD.

4.2.1 Type of PSD Function

The type of statistical function to use is determined by performing digital image analyses on scanning electron microscopy (SEM) images of the external surfaces of the perfusive particles. Frequency data of particle and macropore sizes are obtained from the image analyses and probability plots are made. It will be shown in Chapter 5 that frequency data of particle and macropore sizes correlate very well with the Gaussian distribution. However, at much higher magnifications on the SEM, the images obtained were not sharp enough for image analysis to be performed for the micropores. There is no reason,



- PICK A POINT AT RANDOM
- SELECT AN ORIENTATION FOR PORE
- CHOOSE A DIAMETER
- APPLY PERIODIC BOUNDARY CONDITIONS FOR SURFACE PORES
- USE ONLY DOWNWARD POINTING PORES FOR TOP AND BOTTOM FACES

Figure 4.3 Outline of lattice generation for mapping porous media

nonethelesss, for the distribution function describing the micropore sizes to be different from that describing the macropore sizes. Nevertheless, for the sake of completeness, a sensitivity analysis is performed for the type of statistical function to use for the micropores, namely whether to use a log-normal or Gaussian distribution. For the time being, it suffices to assume that a Gaussian distribution is also used to represent the micropore sizes. Although the PSD used in this work is predominantly the Gaussian distribution function, the network model can be generated using any other statistical distribution functions depending on the actual porous structure.

The PSD used in the network is hence made up of the sum of three Gaussian distributions to represent the three types of pores present in the column (as mentioned earlier):

$$PSD(D) = \sum_{i=1}^{3} \frac{p_i}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{D-\mu_i}{\sigma_i}\right)^2} \qquad ...(4.1)$$

where $PSD(D)$ = pore size distribution,

p_i = relative mathematical proportion of pores of class i,

μ_i = mean diameter of pores of class i,

σ_i = standard deviation of diameter of pores of class i,

D = diameter of pores

4.2.2 Boundary Conditions

In this model, a three-dimensional network is used since only in three-dimensions is it possible to have two continuous phases present simultaneously. In two-dimension networks, one phase can be continuous - the second must be discontinuous. It is the generally accepted view that in actual

porous media, there is a range of saturations where at least a portion of each phase is continuous (Dullien, 1992). Hence, two-dimensional networks cannot be used to accurately model actual porous media. Moreover, with the help of supercomputers like the CRAY Y-MP, highly intensive and complex calculations can be handled even when resorting to three-dimensional models.

At the surfaces of the lattice, periodic boundary conditions are applied for the x- and y-directions. These boundary conditions are necessary so as to take into account end effects and finite size modelling. Figure 4.4. gives a two-dimensional representation of the model used in this work. The two surfaces in the z-direction map the top and bottom surfaces of the finite section of the HPLC column being represented. The remaining surfaces in the orthogonal direction are replaced by the mentioned periodic boundary conditions. The periodic conditions are achieved by translating nodes into each of the eight unit cells (in three-dimensions) surrounding the network to obtain connections as indicated by pores 1-1, 2-2, 3-3 and 4-4 in Figure 4.4.

Thus, in generating the lattice, each time a pore at the periodic surfaces is selected and assigned a pore diameter, the corresponding lattice bond on the opposite face is assigned the same pore diameter in the same pore orientation. As for the z-direction, pores on the top and bottom surfaces are assigned the downward orientation only. The procedure for selecting a pore and assigning a diameter is continued until a predetermined number of pores have been added. The number of pores added determines the pore-interconnectivity (PI) of the porous structure.

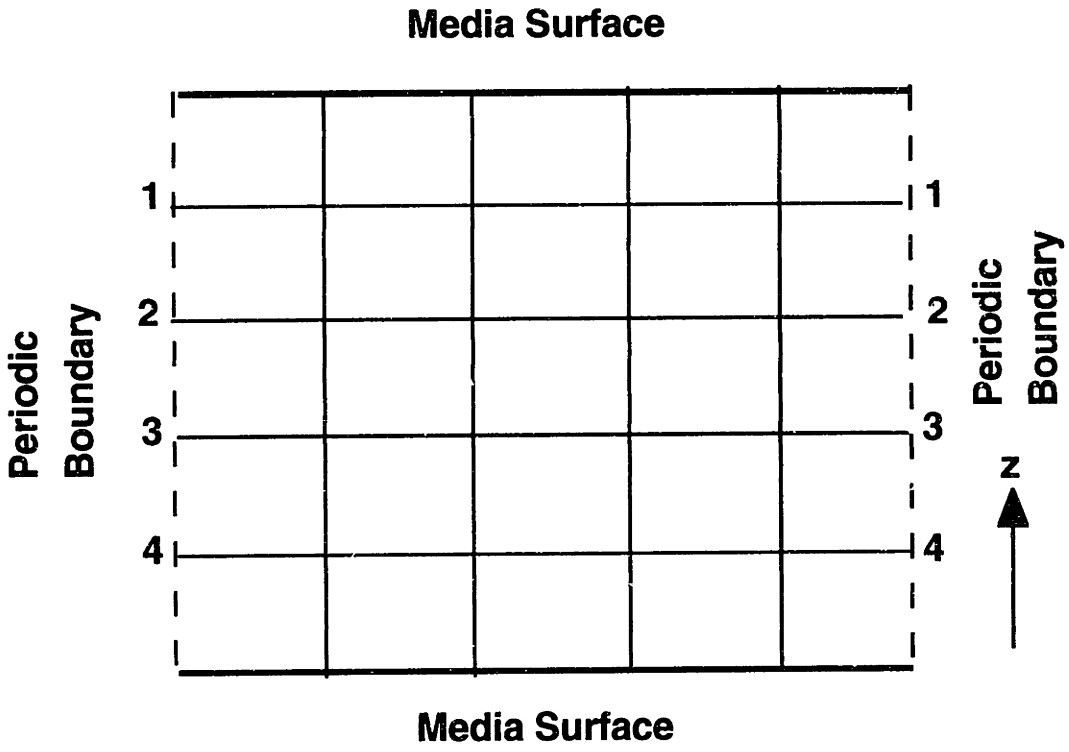


Figure 4.4 Schematic two-dimension representation of the periodic lattice network model

4.2.3 Pore Interconnectivity (PI)

The total number of bonds in the cubic lattice is given by $3L^3$, where L is the lattice size. By predetermining the fraction of this total number of bonds which has to be assigned a pore diameter, the average PI of the network can be varied. Table 4.1 shows the number of pores that have to be added to obtain the various PI's for a lattice size of L=25.

Hence, in forming the lattice network, each time a node and a pore orientation are selected, a pore diameter will only be assigned if the PI of that node is less than or equal to the required PI rather than assigning the predetermined number of pores randomly to the lattice. In this way, local non-isotropicity in terms of pore interconnectivites can be avoided and the average PI in the network will be more uniform. Here, constant network connectivities will be considered, although in general, a connectivity distribution can be specified for the network.

4.2.4 Random Number Generator

In Monte Carlo simulations, the use of an appropriate random number generator is very important in order that the representations are realistic. Here, the pore coordinates and pore orientation are chosen using a non-biased uniform number generator. In the case of pore diameter, a PSD-specific generator (Park and Keith, 1988) is more appropriate. Pore diameters are selected from the corresponding PSD according to the relative probability of abundance of each pore in the Gaussian distribution.

Random numbers are generated from the Gaussian distribution using the inverse cumulative distribution function technique by interpolation of points

Table 4.1 Relationship between fraction of lattice occupied and pore interconnectivity

Fraction of total number of pores	Number of pores for lattice size, L=25	Pore Interconnectivity, PI
0.3	14065	2.2
0.4	18750	2.5
0.5	23440	3.0
0.6	28125	3.6
0.7	32815	4.2
0.8	37500	4.8
0.9	42190	5.1

of the distribution function. The interpolation is achieved using an algorithm attributable to Akima (1970), using piece-wise cubics. For a full description of the algorithm and accuracy comparisons between this method and linear interpolation, the interested reader can refer to Guerra *et. al.*, (1976). Here, it suffices to know that the relative errors using the Akima interpolation are generally considered very good. The random pore diameters used for generating the network is therefore obtained from the subroutines RNGCS and RNGCT in the IMSL library (IMSL, 1991).

4.2.5 Bond Length

The bond length used in setting up the lattice network can be constant, randomly assigned, or related in some manner (directly or inversely proportional) to the diameter. Photomicrographic studies have shown, however, that the length of a pore is of the same order of magnitude as its diameter (Dullien, 1992). Different studies (Imdakm and Sahimi, 1991; Rege and Fogler, 1987; Fatt, 1956; Rose, 1957) have assigned lengths to the bonds by different methods. Although sensitivity analyses (Rege and Fogler, 1988) and the network simulations of porous media made by the abovementioned workers have demonstrated that network model predictions for transport properties based on all the various methods agree well with experimental observations, at the present time, it is unclear as to which of these alternative tends to be most naturally prevalent in porous media. However, it seems likely that larger diameter pores will tend to be associated with larger pores rather than the reverse. In this network model, a pore length equal to the diameter of the pore is used. Topologically, this is a reasonable assignment and by no means is this a limitation of the model. However, in using the network model for fluid transport

analysis, it is necessary to account for the effects of a finite length to diameter ratio on the pressure drop in the pore.

4.2.6 Pore Assignment for Interstitial and Intraparticle Pores

In a column which is formed by the packing of micro-porous beads, there are two distinct classes of void structures; interstitial and intraparticle pores are correlated and therefore cannot be assigned to the network randomly (Hollewand and Gladden, 1992). If pore sizes are assigned to the network from widely differing distributions in an uncorrelated manner, the network generated will be such that there will be pore-shielding of some of the interstitial pores by the smaller intraparticle pores, which will then be an unrealistic representation of the porous structure existing in the column. In a representative section of the column, the interstitial voids form a percolating pore cluster which in turns connects the highly interconnected intraparticle voids. Hence, to generate a realistic network representation of the column, the interstitial voids are assigned in a "semi-random" manner to a percolating cluster which transects the lattice. Although the diameters of the interstitial pores are assigned randomly from the predetermined PSD, the pore position and orientation are partially randomly selected in such a manner that the interstitial pores are connected in a percolating cluster. The interstitial pore assignment is described as follows: A node on the top face of the lattice is selected and an interstitial pore diameter is assigned to the downward pointing pore (recall that all pores on the top and bottom faces of the lattice are assigned downward pore orientation). The determination of a cutoff diameter that segregates the interstitial pores from the intraparticle pores will be discussed in Chapter 5. The downward pointing pore of the node below is next assigned an interstitial pore diameter. This process is continued until the bottom face of the lattice is reached. The remaining

interstitial pore diameters are then assigned to another connecting cluster which branches off the percolating cluster. In this connecting cluster, the pore orientations will be selected randomly. As for the intraparticle pores, assignment of pores in the network is completely random to simulate the isotropicity of the particle pore structure formed from polymerization.

4.3 Mercury Intrusion Simulation

Once the network has been constructed, the process to simulate mercury intrusion is first initiated. By comparing the simulated intrusion curve with the experimental data, the pore size distribution used to generate the network is adjusted until there is a good fit between the two curves. The adjustment is made by systematically changing the parameters which define the PSD. The goodness-of-fit between simulations and experimental data is assessed by a parameter similar to the chi-square factor (Havilcek and Crain, 1988). This will be described later. In this way, the PSD of the perfusive particles is inferred through an iterative trial and error computational procedure. In addition, matching the simulated and experimental curves will incorporate the relevant porosities inherent in the actual porous structure in the network model. Mercury intrusion on the network is simulated using the algorithm shown in Figure 4.5 and is modelled after Lane (Lane, 1983). Initially, the void space is assumed to be evacuated. The program starts by incrementally increasing the pressure, emulating the actual experimental procedure. In mercury porosimetry, the Washburn's equation (Equation (4.2)) is used to relate the intrusion pressure to the diameter of a cylindrical pore:

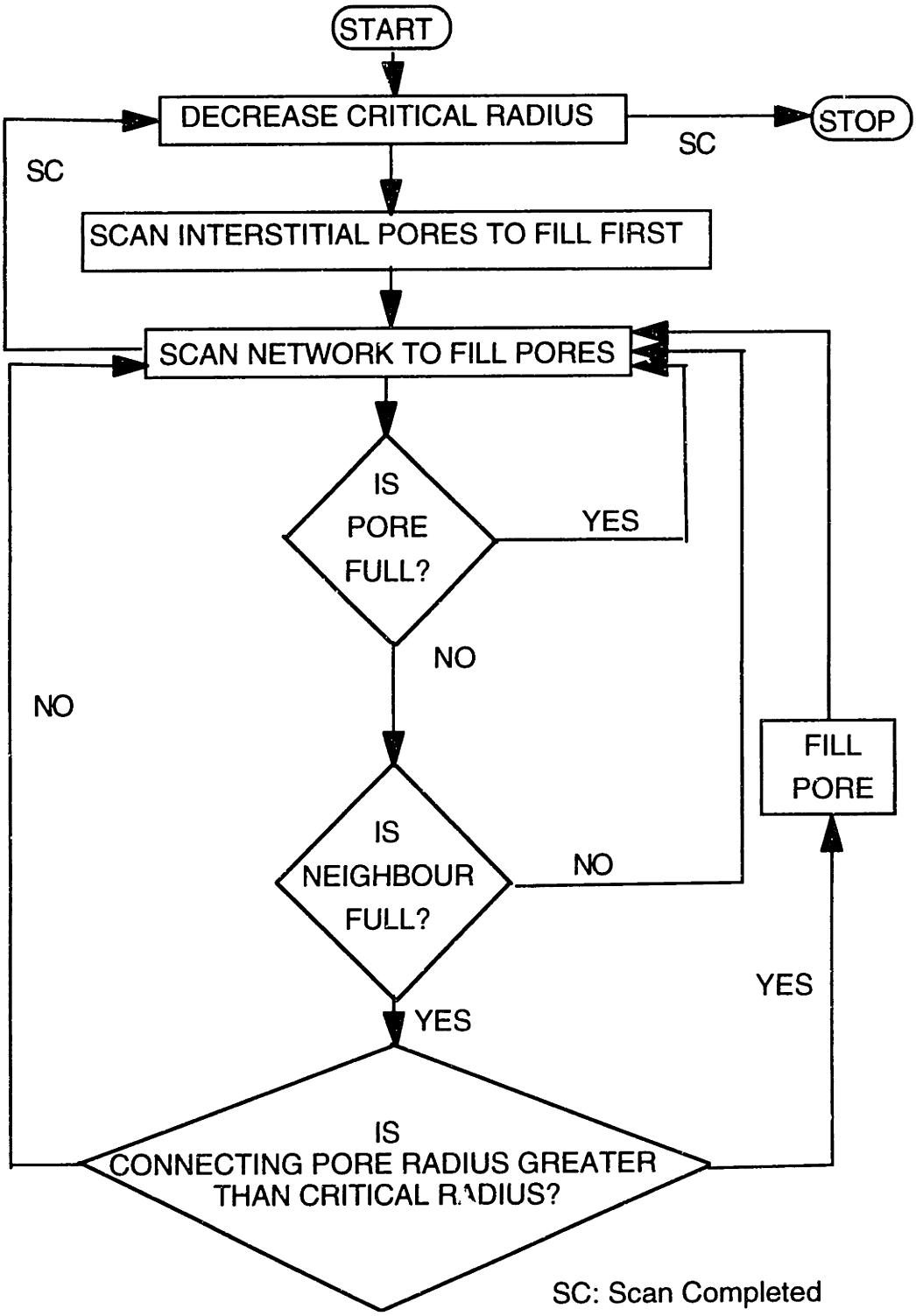


Figure 4.5 Flow chart for mercury porosimetry simulation

$$P = \frac{-4\gamma\cos\theta}{D} \qquad \ldots(4.2)$$

where P= intrusion pressure (psia)

γ= surface tension of mercury (485 dynes/cm)

θ= contact angle between mercury and pore surface (140^{o})

D= pore diameter (Å)

Incrementally increasing the pressure can be accomplished by the equivalent process of decreasing the critical diameter for intrusion according to Equation (4.2). The first pores to be filled are the interstitial pores in the percolating interstitial cluster. These provide the mercury menisci necessary for mercury penetration into the microporous structure of the beads.

The network is scanned to intrude pores based on a number of criteria. A pore will be filled if (1) the pore is empty, (2) at least one of the pore's neighbours is filled, and (3) the diameter of the empty pore is at least equal to the current critical pore diameter. Upon satisfying the above conditions, the pore is filled with mercury and the network is scanned again at the same critical diameter until no further pores can be filled. The cumulative volume of mercury that has intruded into the void space at that pressure (critical diameter) is then recorded. The critical diameter is then gradually decreased and the intrusion process repeated until the size of the smallest pore in the network is reached.

Since the network model is used to represent a finitely small section of the column, several independent realizations of the system have to be generated. The simulated intrusion curve obtained from a single realization will be " jagged" in character and is unique to the particular set of random numbers

used in generating the model. A realization is defined as the different networks generated using different selection of random numbers to account for statistical variations. Different sets of random numbers can be generated using different initial seeds for the random number generator. The number of realizations needed usually depends on the size of the sample, and on the desired accuracy. For large samples, a few realizations often provide accurate estimates of the quantities of interest.

To determine the number of realizations required, it is necessary to obtain the stochastically averaged simulated curve over a fixed number of realizations. Over the intrusion pressure range used in the porosimetry experiment, an average of about 15 pressure points are selected. For each mercury intrusion experiment simulated for each realization, the cumulative intrusion volume at these pressure points are recorded. After a fixed number of realizations have been carried out, the arithmetic mean and the standard deviation of the cumulative intrusion volumes at each of the pressure points are calculated. The required number of realizations are obtained when the standard deviation at each of the selected pressure points is less than 5% from the stochastically averaged simulated curve.

In matching the average simulated and experimental intrusion curves, the sum of squares of the residuals, given by Equation (4.3), is used to test the goodness-of-fit:

$$Goodness-of-fit, F = \sum_i \left(\frac{y_{isim} - y_{iexpt}}{y_{iexpt}}\right)^2 \qquad ...(4.3)$$

where y_{isim} is the pressure at simulated intrusion volume V_i, and

y_{iexpt} is the experimental pressure at the same intrusion volume V_i

The F factor defined in Equation (4.3) is similar to the chi-square (χ^2) factor used in statistical analysis (Havilcek and Crain, 1988). Note that the lower the F factor, the better is the fit between simulation and experimental data. The pressures were used in the F factor since the pressures vary in a logarithmic fashion while the intrusion volume varies linearly. Consequently, minimizing the errors in the pressure will inherently minimize the errors in the intrusion volume.

The goodness-of-fit, F factor, gives the total fractional error of the closeness of fit between the simulated and the experimental curves and the objective is to adjust the inputs of the lattice network (parameters in the PSD and the PI) such that a minimum F is achieved.

4.4 Adsorption Capacity Simulation

It can be shown and derived from principles of statistical mechanics, that the exclusion probability, E(r), of a solute in a pore depends on the size and shape of the solute molecules and on the size and shape of the pores in the column packing material (Giddings *et. al.,* 1968). For the case of a solute molecule which can be taken as a "hard sphere" of radius r and the pore an infinite cylinder of radius R, it follows from steric interactions (Knox and Scott, 1984) that the center of mass of the molecule cannot approach any closer than a distance r from the wall of the pore and the part of the pore volume which is accessible to the center of mass of the solute is a cylinder of radius (R-r).

As a consequence, the exclusion probability, which is the probability that a pore is accessible to the molecule can be presented as:

$$E(r) = (1 - \frac{r}{R})^2 \qquad (\frac{r}{R}) < 1$$

and $$E(r) = 0 \qquad (\frac{r}{R}) > 1$$

Based on this relationship, the accessibility of a protein molecule in a cylindrical pore can be assessed in the network model to simulate the adsorption capacity of the perfusive particles for different proteins. For the lattice network, the total accessible area for adsorption can be obtained from:

$$A(r) = \sum_{R=r}^{R=\infty} (1 - \frac{r}{R})^2 4\pi r^2 \qquad ...(4.4)$$

since the pore length is the same as the pore diameter in the network.

Using the same algorithm as that used in simulating mercury intrusion, with the critical diameter representing the diameter of the protein, the process of protein adsorption is simulated on the network model. Here, each time a pore is accessed, the total accessible area is calculated from Equation (4.4) to represent the adsorption capacity.

4.5 Size Exclusion Chromatography Simulation

Size exclusion chromatography (SEC) has been mainly employed as an analytical procedure for separating macromolecules according to differences in size. While simulating protein adsorption on the network model predicts experimental data based on the surface area of the packing beads, simulating SEC will elucidate experimental results based on the volumetric porosity of the particles. Different models have been proposed to evaluate the partition

coefficient, K, of flexible-coil macromolecules between the bulk solution and cylindrical pores (Giddings *et. al.,* 1968; Casassa, 1972; Davidson *et. al.,* 1987). By using a Monte Carlo simulation technique, Davidson and his coworkers (1987) provided a simple and most complete relationship which relates the partition coefficient to the radius of gyration of a flexible- coil macromolecule:

$$\ln K = \ln K_o + (l/R)(0.49 + 1.09\lambda_G + 1.79\lambda_G^2) \qquad ...(4.5)$$

where K = partition coefficient

$K_o = 4\sum_{i=1}^{\infty}(\frac{1}{\alpha_i^2})\exp(-\alpha_i^2\lambda_G^2)$

$\lambda_G = \frac{r_g}{R}$ = ratio of radius of gyration of macromolecule to pore radius

α_i = roots of the Bessel function of the first kind and of order zero

l = segment length of the macromolecule, given by:

$r_g = l\sqrt{\frac{n^2-1}{6n}}$ where n is the number of segments

For the lattice network, the partition coefficient for different polystyrene molecules can therefore be obtained as:

$$K(r_g) = \sum_{R=r_m}^{R=\infty} F(R)K(r_m)$$

where $F(R)$ = accessible volume fraction of pores in the network taken up in radii R.

In this manner, size exclusion chromatography is simulated on the network model using the same algorithm as that used in simulating mercury intrusion. In this case, the critical diameter represents the diameter of gyration of the polystyrene molecule.

4.6 Medium Permeability

An important application of the network model developed is the quantification of intraparticle convection in perfusive media. The goal is to determine the permeability of the media to liquid flow relative to total flow through the bed. This permeability ratio translates directly to the ratio of liquid velocity in the media to superficial velocity in the HPLC bed. This, concomitantly, relates to column efficiency via a macroscopic modelling analysis developed by Rodriguez and his co-workers (Rodriguez *et. al.*, 1991; 1992; 1993). The ultimate objective is to illustrate the influence of the parameters describing the pore size distribution and pore interconnectivity on column efficiency. In this section, the transport methodology necessary to elucidate the permeability ratio will be described.

A fixed flow rate is imposed on the network and the background pressure at the nodes of the lattice is determined. The limit of Reynolds number of flow in the pores is first calculated to ascertain the flow regime in the media. This is useful in determining the entrance length effects in the pores for fully developed Poiseuille flow.

Considering the largest pore in the network to be the size of the packing particle, that is, d_p= 20μm, at a flow rate of u = 5000 cm/h,

$$\mathrm{Re}_{max} = \frac{d_p u \rho_l}{\mu}$$

$$= \frac{(20x10^{-4})(\frac{5000}{3600})(1)}{0.01}$$

$$= 0.278 < 1$$

Hence, creeping flow exists in all the pores of the network and the entrance length for fully developed flow becomes independent of Re, that is,

$$\frac{L_e}{D} = 0.59 \quad \text{(Atkinson } et.\ al.\text{, 1969)} \qquad \ldots(4.6)$$

where L_e is the entrance length, and

D is the pore diameter.

For fully developed flow, however, it is necessary that $L_e >> 10D$ (Deen, 1994). Since the length of the pores in the network model has been taken to be equal to the pore diameter, the Poiseuille flow equation is inadequate for describing pressure drop in the short pores.

Dagan *et. al.* (1982) have determined a correlation for the pressure drop through a cylindrical pore of finite length under creeping flow conditions by matching the normal and tangential components of the viscous-stress tensor at the openings of the pore. They have found that the actual flow field through a short pore deviates from the Poiseuille flow only in the vicinity of the pore openings and even in these regions, the local axial velocity profile does not depart significantly from a parabolic distribution. The entrance effects are significant only near the pore openings and decay to within 1.5% of a Poiseuille profile at a distance of half the pore radius when L/D > 0.5. The pressure drop across a short pore of finite length can be closely approximated (to less than 1%

in error) by Equation (4.7), which does not need to account for the upstream influence across the pore opening.

$$\Delta P = (\frac{16 l_p / D}{\pi} + 3)\frac{8Q\mu}{D^3} \qquad ...(4.7)$$

where ΔP is pressure drop across the pore,

l_p is the pore length,

D is the pore diameter,

Q is flow rate through the pore,

μ is viscosity of the liquid

By rearranging Equation (4.7), the flow rate through each pore in the network can be obtained as:

$$Q_{ij} = \frac{(P_i - P_j)D^3}{(\frac{128}{\pi} + 24)\mu} \qquad ...(4.8)$$

whereby a pore is connected through nodes i and j.

At the internal nodes of the lattice, there will be no accumulation and zero net mass flux of fluid elements at these results in a set of linear equations for the nodal presures. This set of equations form a banded matrix since the material balance only accounts for the six nodes surrounding the node in question. The banded matrix is then solved by an L-U decomposition technique. This is performed using the IMSL subroutines, SGBTRF and SGBTRS.

In forming the banded matrix, it is necessary to remove nodes in the lattice which have either no pores attached to them or the pores attached to them are dead-ended pores. These, if not properly eliminated, will result in

singularity of the matrix and there will be convergence problems associated with the L-U decomposition. The reason for the presence of nodes which have zero connectivities is due to the fact that the lattice network is not completely occupied at PI's less than the coordination number of the cubic lattice (Z=6). These nodes, which need to be removed, can be identified by locating the interconnected pore cluster (thereafter termed the infinite cluster, in accordance with the terminology of percolation theory) in the network. In this way, any node which does not belong to the infinite cluster will be nodes of zero PI and will be discarded. A scan of the infinite cluster for nodes with PI = 1 will then provide the dead-ended nodes which also have to be removed. The algorithm used for locating the infinite cluster is based on the cluster multiple labelling technique first proposed by Hoshen and Kopelman (1976). The success of the method, and its versatility in solving complex percolation problems, is based on the application of alternate labels to sites (nodes) belonging to the same cluster. The end result of this is that site labelling and cluster size classification can be accomplished by only a single scan of the network which concomitantly reduces computer storage requirements. The computer listing detailing the algorithm for use in this research can be found in the Appendix (program name: HOSHEN.f). For a complete description of the technique, the interested reader is referred to the paper by Hoshen and Kopelman (1976).

After the background pressure is obtained, the intraparticle convective flow throught the perfusive media is calculated relative to the total flow through the bed. Recall that the interstitial pore assignment is segregated from the allocation of intraparticle pores in the network; the interstitial pore cluster percolates the network. From the solution of the background pressures, the pressure drop through this interstitial pore cluster is calculated. This represents

the amount of flow through the interstitial pores which can then be translated directly to the amount of intraparticle flow through the perfusive media by subtraction. From this, the fraction of the total flow through the bed which is attributed to intraparticle flow can be obtained. Converting this fractional intraparticle flow rate to velocity ratio in the particle becomes straightforward:

$$Q_p = u_p S(1-\varepsilon_b) \qquad \text{...(4.9)}$$

$$Q_b = u_b S \qquad \text{.(4.10)}$$

where Q_p is the intraparticle flow rate

Q_b is the total flow rate through the bed

u_p is the intraparticle superficial velocity

u_b is the superficial bed velocity

S is the column cross-sectional area

ε_p is the particle porosity

ε_b is the bed porosity

Therefore, $$\frac{u_p}{u_b} = \frac{Q_p}{Q_b}\frac{1}{(1-\varepsilon_b)} \qquad \text{.(4.11)}$$

Subsequently, this velocity ratio (Equation 4.11) is directly related to column efficiency using macroscopic modelling methodology developed by Rodriguez and his co-workers. This subject will be taken up in detail in Chapter 5.

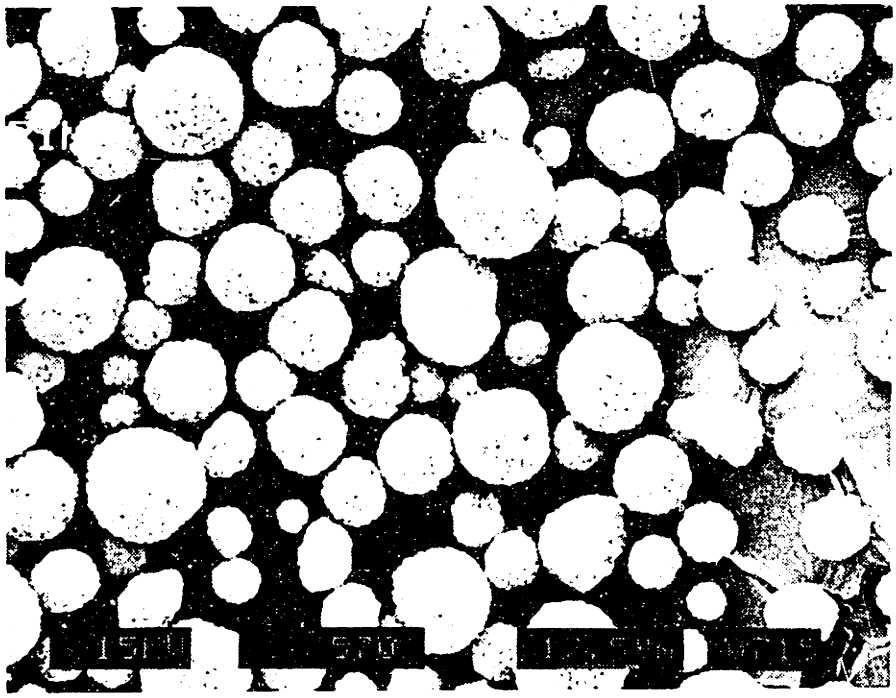
Chapter 5 -

Results and Discussion

5.1 Scanning Electron Microscopy and Digital Image Analysis

Scanning electron microscopy (SEM) images of the external surfaces of the perfusive materials were taken to determine the mathematical form of the statistical function to use for the pore size distribution (PSD) in the network model. Since it has been reported (Unger and Gimpel, 1979; Colin and Guichon, 1976; Tracz *et. al.,* 1984; 1986) that the overall PSD of the particle is correlated to the porous structure of the external surface, images of the fractured particles are not required. The SEM measurements were taken at 500x and 10,000x magnifications and five frames of each magnification were obtained. Figure 5.1 shows the typical SEM images of POROS I and POROS II particles while Figure 5.2 shows the typical SEM images of the external surfaces of the particles. These images were digitized and analyzed to obtain the frequency data of the particle sizes (Figure 5.1) and macropore sizes (Figure 5.2) of the packing materials. It is important to point out that there is a distribution in the

(a)



(b)

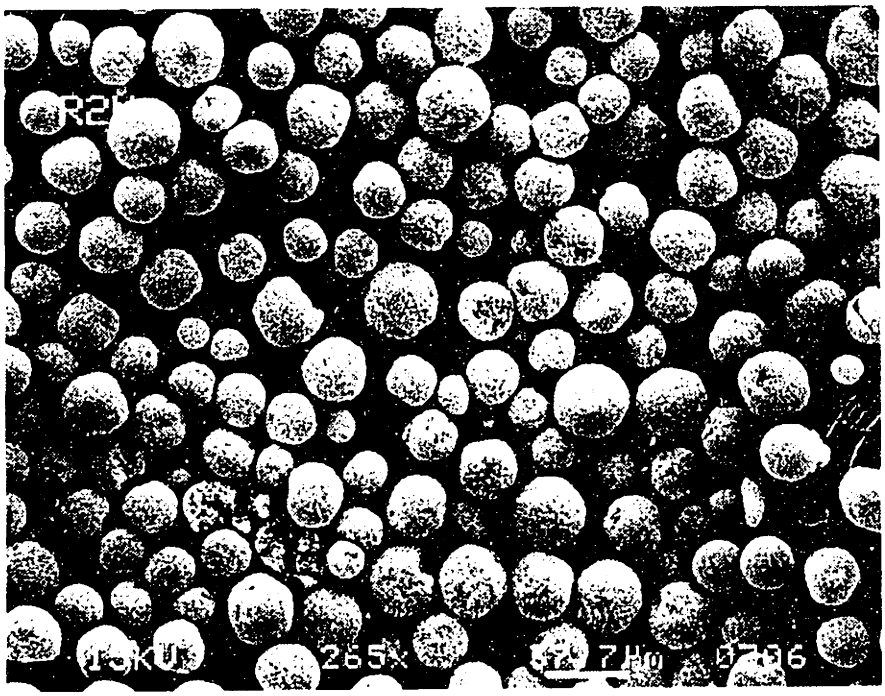
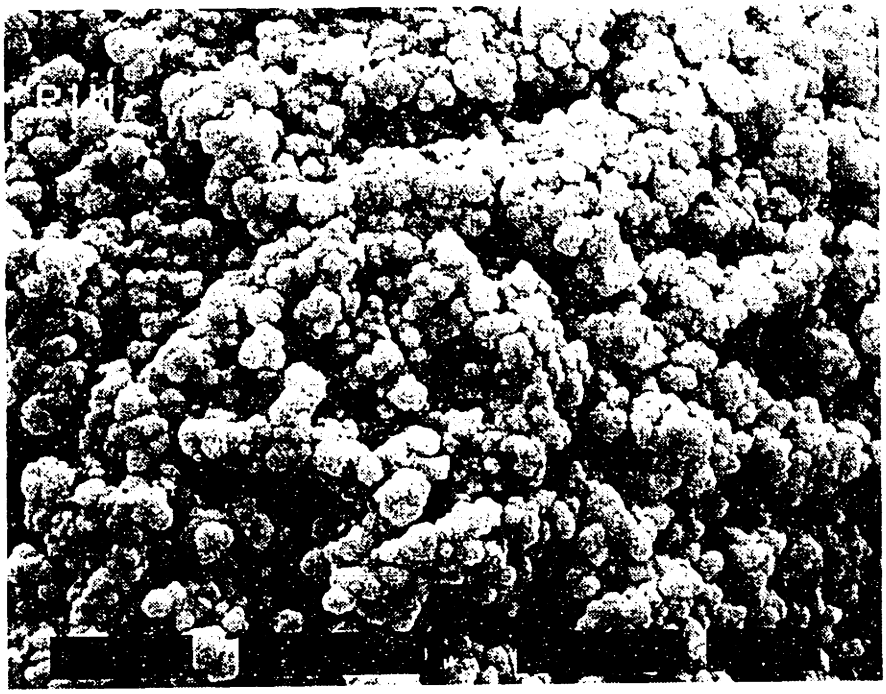


Figure 5.1 Typical SEM images of POROS particles.

(a) POROS I at 570x magnification (b) POROS II at 270x magnification

(a)



(b)

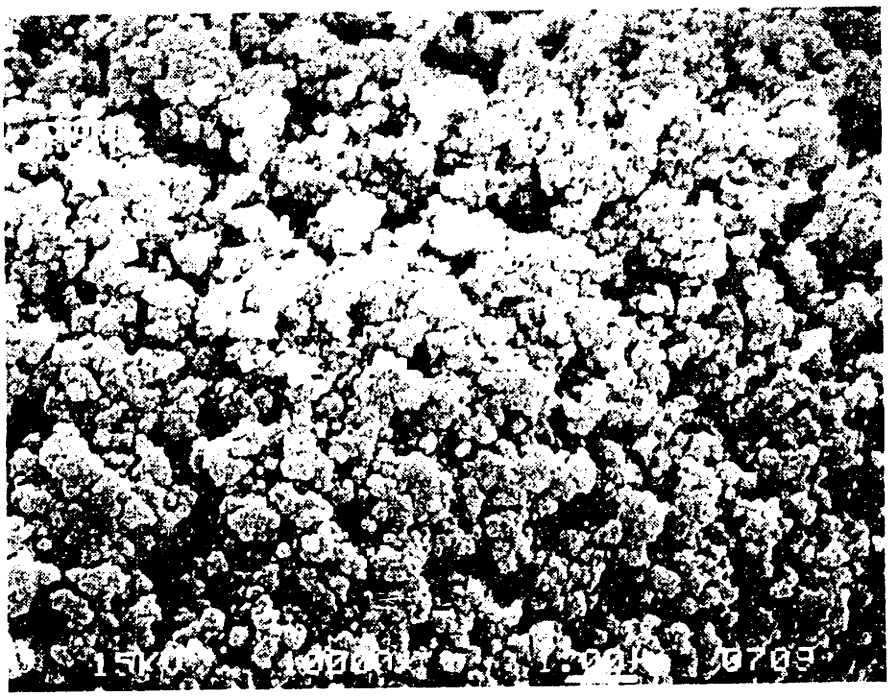


Figure 5.2 Typical SEM images of the external surfaces of POROS particles at 10,000x magnification. (a) POROS I (b) POROS II

particle sizes of the packing media and the presence of fine particles as evident in Figure 5.1. In both cases of POROS I and II, there exists a small number of particles with diameters in the range of 10-14μm. The possible effects of these fine particles on the column porosity will be discussed in more details in a later section.

Figure 5.3 plots the histograms of the particle size distributions obtained from the image analyses. In the case of POROS I, a total of 400 particles were counted while a total of 700 particles were counted for POROS II. Based on the frequency data, the volumetric average particle diameter was calculated. The volumetric average was necessary for comparison with the average particle size obtained from laser scattering experiments. From image analyses, the average particle diameters were found to be 22 μm for POROS I and 23 μm for POROS II. These values compare favorably with the 23 μm average for both POROS I and POROS II from laser scattering techniques. To determine the type of particle size distribution, probablity plots of the frequency data were made. Shown in Figure 5.4(a) and (b) are the probability plots for the particle size distributions of POROS I and II respectively. Also shown in the graphs are the linearly-regressed lines of the data. The data are clearly linear and correlation coefficients of r^2=0.96 were obtained in both cases. The closeness of fit of the frequency data to linearity reveal that the particle size distributions of POROS I and II follow the Gaussian distribution. To confirm this, the skewness factor, S_k, defined by Equation (5.1) was calculated for the two sets of data presented in Figure 5.3.

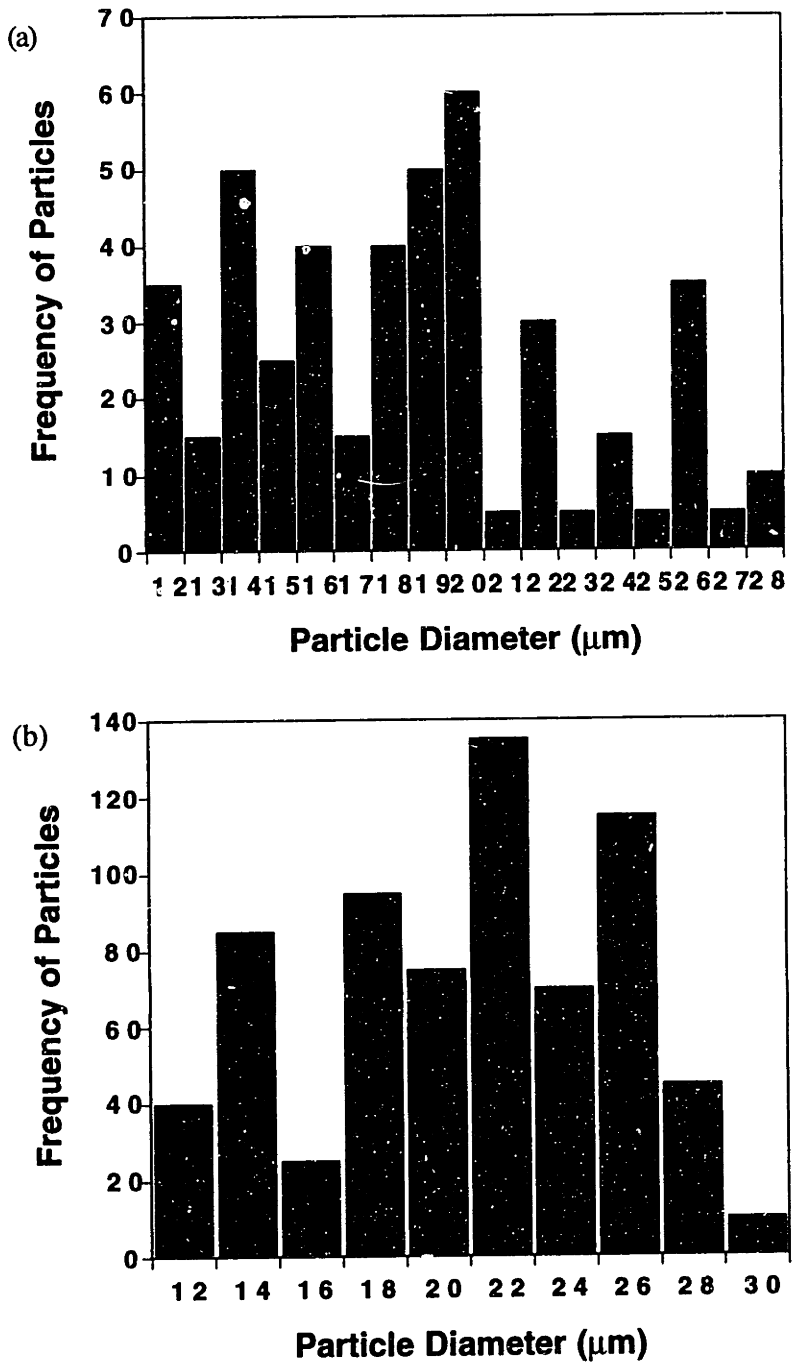


Figure 5.3 Frequency data of particle diameter obtained from image analysis on SEM images of (a) POROS I and (b) POROS II.

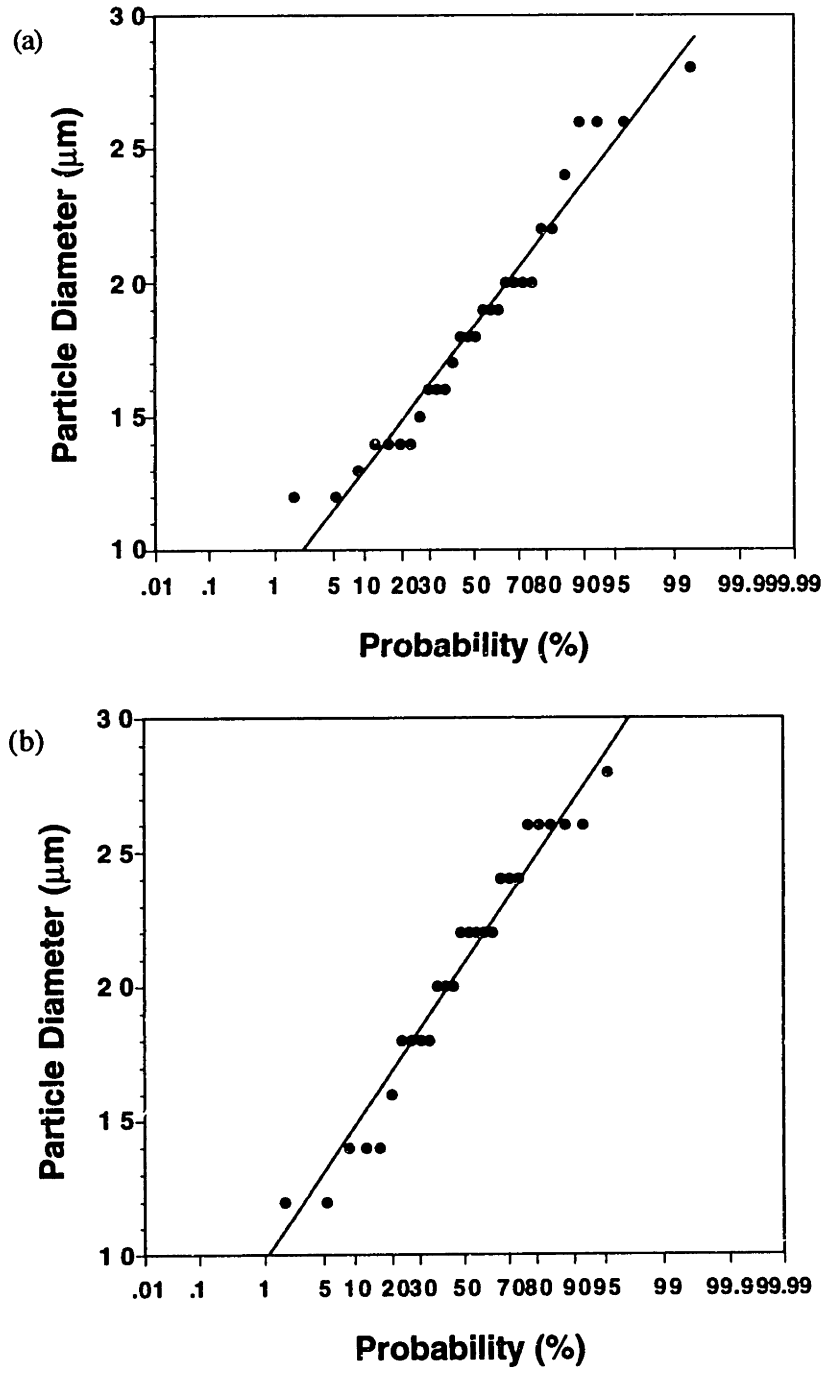


Figure 5.4 Probability plot of particle size distribution of (a) POROS I and (b) POROS II. The regressed lines are also shown in the graphs. The correlation coefficients are r^2=0.96 in both cases.

$$S_k = \frac{\sum_i (D_i - \overline{D})^3}{n[\frac{1}{n-1}\sum_i (D_i - \overline{D})^2]^{\frac{3}{2}}} \qquad ...(5.1)$$

where D_i is the particle diameter

$\overline{D}$ is the mean diameter

n is the total number of particles counted

S_k is a quantitative measure of how close the frequency data approaches a Gaussian distribution. For a perfect Gaussian distribution, S_k should be equal to zero. Based on Monte Carlo probability points computed by D' Agostino (1971), the 95% confidence intervals for the skewness factor were obtained as:

$-0.522 < S_k < 0.522$ for POROS I

$-0.400 < S_k < 0.400$ for POROS II

The skewness factors calculated from the frequency data were 0.44 and -0.27 for POROS I and II respectively. These clearly fall within the 95% confidence interval which further supports a Gaussian distribution for the particle size distribution for the perfusive packings.

Typical SEM images for the macropores are shown in Figure 5.2(a) for POROS I and Figure 5.2(b) for POROS II. It can be seen from these images that it is very difficult to define a pore. The dark areas which are indicative of open structures on the surfaces were taken to be the pores and the diameter of the equivalent circle of these regions were obtained from image analysis. Figure 5.5 plots the histograms of the resulting macropore size distribution. A total of

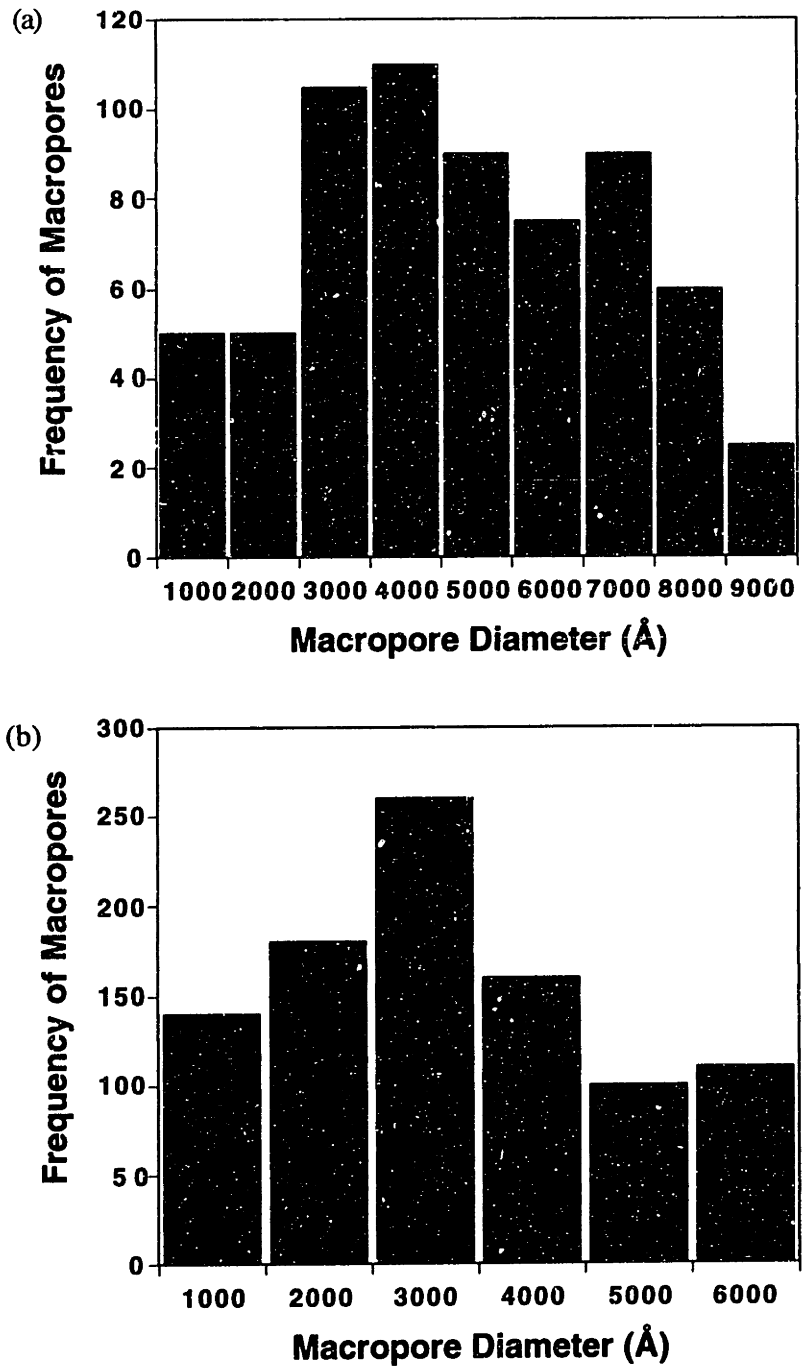


Figure 5.5 Frequency data of macropore diameter obtained from image analysis on SEM images of the external surfaces of (a) POROS I and (b) POROS II.

650 pores were analyzed for POROS I and 1050 pores were analyzed for POROS II. The average macropore size was calculated for these data assuming the pores are cylindrical with an aspect ratio (length to diameter ratio) of one. The average macropore sizes were found to be 5000Å and 3500Å, both significant to 500Å, for POROS I and II respectively. These values will be compared with those obtained from mercury porosimetry simulation using the network model in a later section.

Probability plots of the macropore frequency data are shown in Figure 5.6. Linear regression of the data indicates that the probability plots are linear with correlation coefficient of r^2=0.96 for POROS I and r^2=0.94 for POROS II. Hence, Gaussian distributions are indicated for the macrcpore size distribution of the perfusive media. From the frequency data, the skewness factors were evaluated to be 0.07 and 0.29 for POROS I and II respectively. These values are within the 95% confidence intervals for S_k:

$$-0.414 < S_k < 0.414 \qquad \text{for POROS I}$$

$$-0.328 < S_k < 0.328 \qquad \text{for POROS II}$$

Consequently, it can be ascertained that the macropore size distributions are of the Gaussian type for the perfusive materials. Unfortunately, the same analysis cannot be performed for the micropore size distributions for reasons mentioned in Chapter 3. Nevertheless, there is no reason for the distribution function describing the micropore sizes to be different from that describing the macropore sizes. In Section 5.6, a sensitivity analysis will be presented to compare the use of a lognormal distribution for the micropore size distribution as opposed to using a Gaussian distribution.

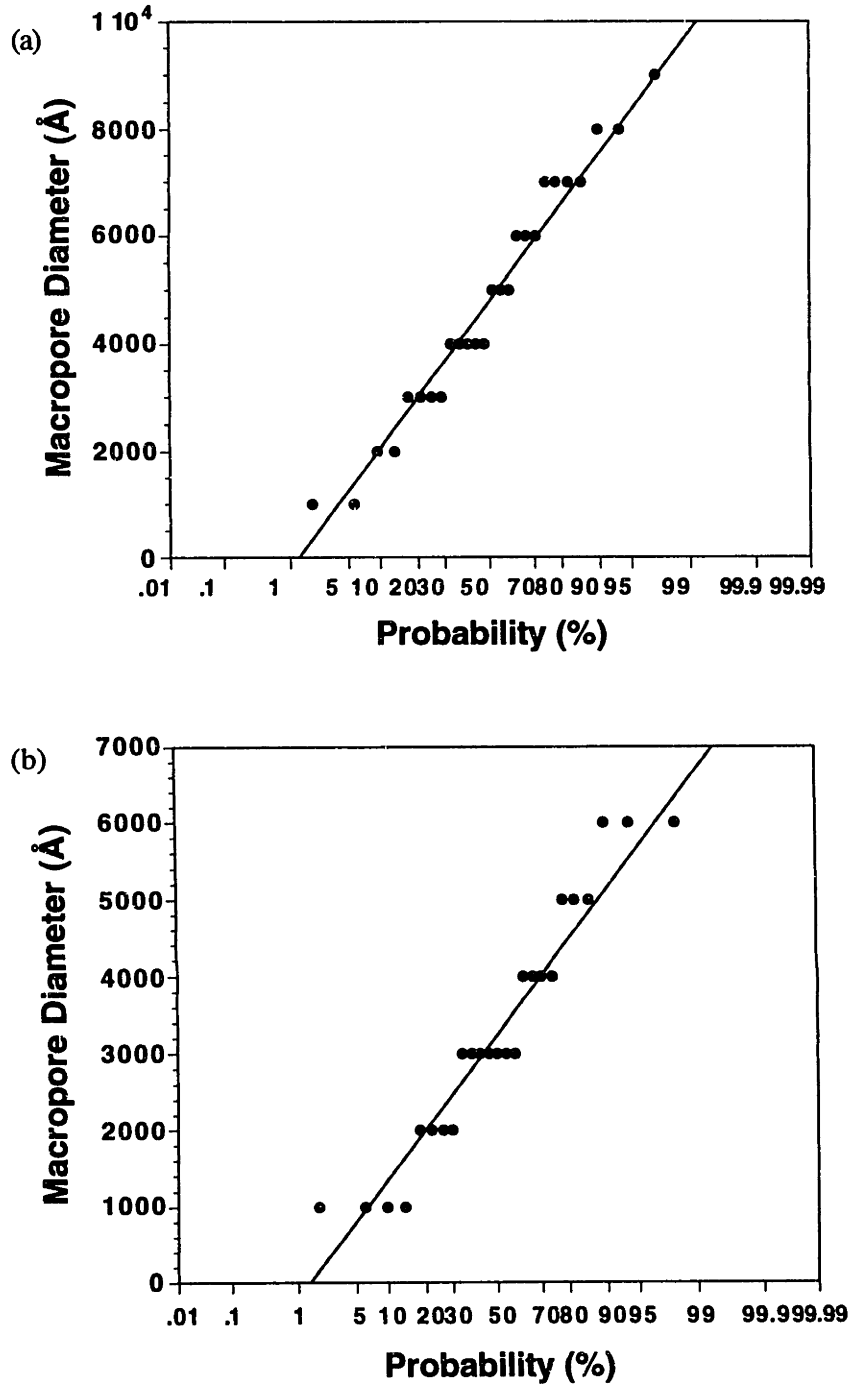


Figure 5.6 Probability plot of macropore size distribution of (a) POROS I and (b) POROS II. The regressed lines are also shown in the graphs. The correlation coefficients are r^2=0.96 for panel (a) and r^2=0.94 for panel (b).

5.2 Mercury Intrusion of POROS I and POROS II

It is necessary to determine the transition between interstitial and intraparticle pore sizes in the simulation of mercury intrusion for the network model. This is because the interstitial pores provide the mercury interface necessary for mercury penetration into the network.

The transition from interstitial to intraparticle pore size was determined from a comparison of the intrusion curves of POROS I and POROS II (Figure 5.7). From Figure 5.7, it can be seen that the departure of the intraparticle pores between the two media occur at approximately 1.0μm. As a result, 1.0μm was taken to be the cutoff between interstitial and intraparticle pores.

From this, the porosity of the packing was calculated as follows:

The porosity of the particle can be related to the volume of mercury intruded by:

$$\varepsilon_p = 1 - \left(\frac{1}{1 + V_{int}\rho}\right) \qquad \text{...(5.2)}$$

where V_{int} = specific volume of mercury intruded into the beads, and

ρ = skeletal density of bead particles (1.05 ml/g)

Using acetone injection, the total porosity of the column, ε_t, was obtained and from

$$\varepsilon_t = (1 - \varepsilon_c)\varepsilon_p + \varepsilon_c \qquad \text{...(5.3)}$$

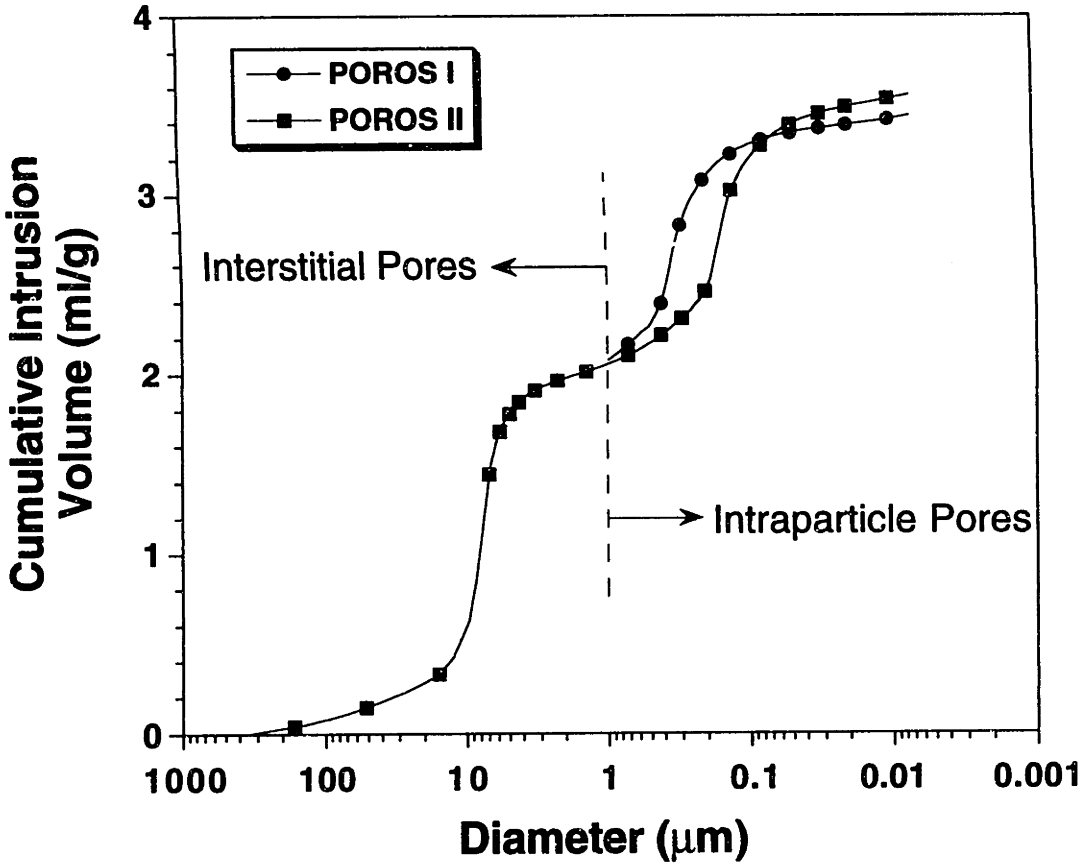


Figure 5.7 Relationship between pore diameter to cumulative intrusion volume for POROS I and II obtained from mercury porosimetry. Dotted line indicates cutoff point between interstitial and intraparticle pores.

the column porosity, ε_c, was calculated. The summarized results of the calculations for POROS I and POROS II are tabulated in Table 5.1.

Table 5.1 Summary of Porosities and Mercury Intrusion Volumes for POROS I and II Columns

Parameter	POROS I	POROS II
ε_c, Column porosity	0.25	0.23
ε_t, Total porosity of column	0.68	0.70
ε_p, Particle porosity	0.57	0.61
V_{int}, Specific mercury intrusion volume (ml/g)	1.26	1.49

The column porosity obtained from the back calculation seems rather low at 25% and 23% for the POROS I and II columns respectively. Column porosities reported for HPLC packings have usually been in the range of 30-35% (Afeyan *et.al.*, 1990; Rodrigues *et.al.*, 1991). For random close packing of spheres of uniform diameter, it was found (Haughey and Beveridge, 1969) that the porosity of the "densest" regular packing exists in the rhombohedral packings, where the coordinate number between the spheres is twelve and the bulk porosity was found to be 25%. Therefore, the low column porosity values obtained from the calculations are at the theoretical minimum for uniform sphere packings.

There are a couple of probable reasons for the discrepancy in the column porosity. The discrepancy may be due to the presence of fine particles in the packing media used to pack the columns. As noted in Section 5.1, there exists a small fraction of the packing particles in the diameter range of 10-14μm. It was shown by Horsfield (1934) that the filling of the voids of a rhombohedral packing with spheres of different and much smaller diameters can result in a bulk packing porosity of 15%. He had noted that while the smaller particles, on the one hand, tend to increase the porosity by forcing the larger particles apart, they can also decrease the bulk porosity by filling the interstitial voids between the larger particles, the latter predominating for a size ratio greater than about 3:1. The reduction in the porosity will, however, not only depend on the size ratio but also on the amount of each size fraction present. From the histogram of particle size frequencies (Figure 5.3), the fraction of fine particles present is small and hence may only play a small role in reducing the column porosity.

A more probable reason for the low column porosities obtained may be due to the experimental error incurred in the acetone injection analysis for obtaining the total porosity of the column. The experimental error in estimating the total porosity of the column was found to be about 10% (about 9% error in measuring the flow rate of the acetone through the column and less than 1% error in timing the elution of the acetone). From Equation (5.3), it can easily be shown that a 3% increase (POROS I) in the total porosity of the column from 0.68 to 0.70 can result in a 20% increase in the column porosity from 0.25 to 0.30 while keeping the particle porosity at 0.57. Therefore, the calculation of the column porosity may be in error resulting from the experimental error incurred in

the total porosity of the column. Unfortunately, it is not possible to verify the column porosity for POROS packings due to the inability to find a solute probe big enough to probe only the interstitial pores without entering the intraparticle pores (Kalghatgi, 1994; McCoy, 1995). In light of the work presented in this thesis, however, it will be shown in Section 5.10.3 that the sensitivity of intraparticle convection to the column porosity is very small and the discrepancy is therefore not detrimental.

The computer simulations were performed on the CRAY Y-MP supercomputer which takes approximately 300 seconds to perform each simulation of the mercury porosimetry experiment. Parameters of the PSD and PI were first obtained for POROS I from the simulations. Since the particle size distributions for POROS I and II are essentially the same (SEM and image analyses data in Section 5.1), the same distribution function describing the interstitial pores in POROS I should apply to POROS II. Moreover, it can be taken that the PI of the porous structure comprising POROS I to be the same as the PI of the porous structure comprising POROS II since the two packings were manufactured by the same processes. In the case of POROS II, therefore, there are four less parameters in the PSD function to be inferred from the mercury porosimetry simulations.

The major disadvantage of using the Gaussian distribution to represent PSD is the generation of negative pore diameters in its use. To overcome this, the algorithm used in the random number generation of pore diameters truncates the Gaussian distribution for negative diameters. This was found to be non-restrictive on the simulations since only less than 1% of the diameters

generated were negative in the case of simulating POROS I and a corresponding value of less than 2% in the case of simulating POROS II.

Figures 5.8(a) and 5.8(b) show a comparison of the intruded volume of mercury as a function of pressure for the simulation and experimental data for POROS I and II, respectively. The experimental data shown are the average of four mercury porosimetry experiments. The average experimental error was found to be less than 5% for both POROS I and POROS II. A total of 92 data points were taken during each experiment.

By adjusting the PSD and PI used in generating the network, the experimental and simulation data were matched to a goodness-of-fit F factor of less than 1.00 for both POROS I and II. This F factor corresponds to an average fractional error of less than 10% for the 92 experimental data points. The only exception in agreement between the simulated and experimental data is a very short segment at the early stage of the experimental intrusion curve which represents the filling of the void space in the sample holder during the porosimetry experiment. This part of the curve is therefore not representative of pore space existing in the HPLC column.

A summary of the parameters for the PSD used in the simulations is tabulated in Table 5.2, and the relative probability distributions are shown in Figure 5.9. From the simulations, it was found that the porous structure of perfusive particles, POROS I and II indeed consists of 2 groups of widely differing pore sizes; a group of large pores (of the order of thousands of angstroms in sizes) through which convection is believed to take place, interconnected by a group of small pores (of the order of hundreds of angstroms

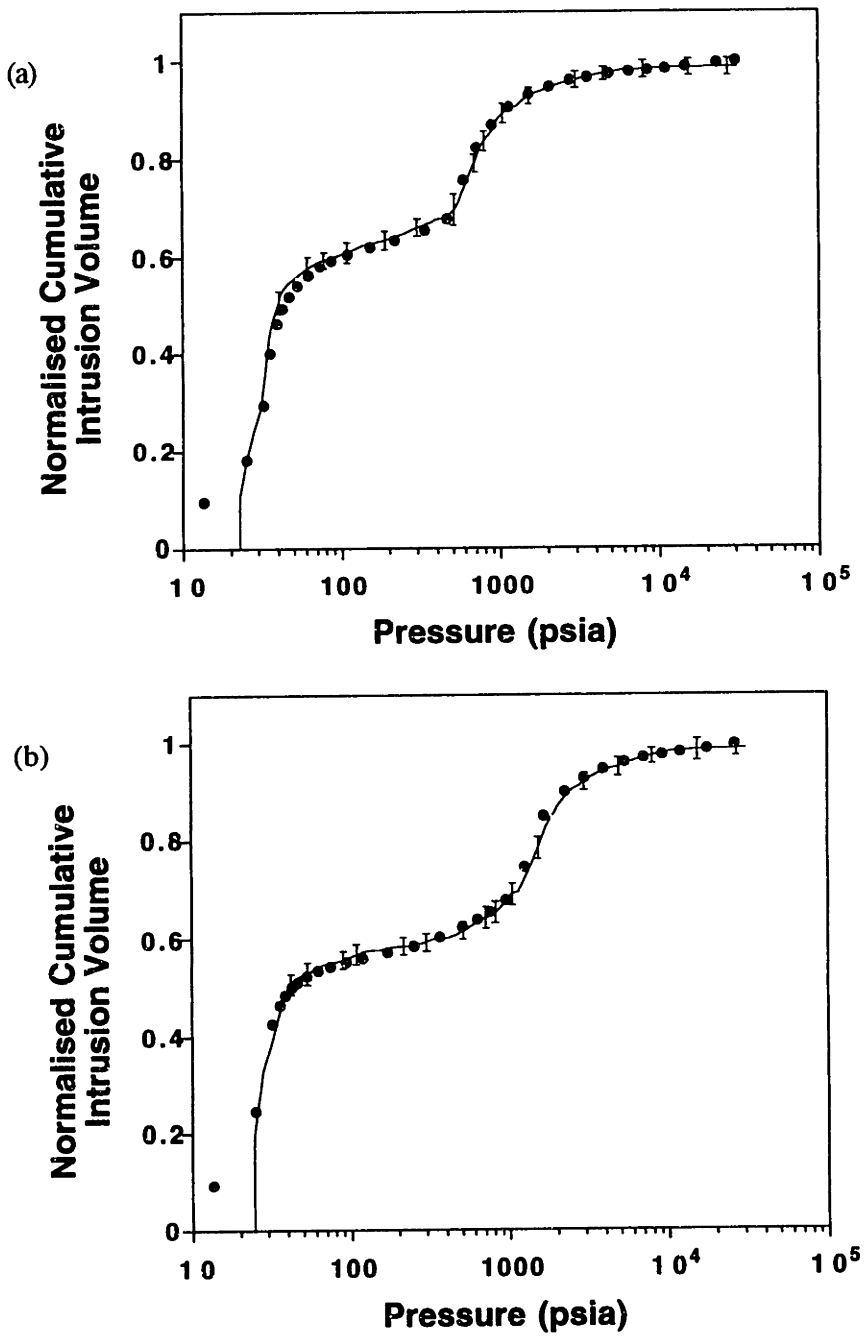


Figure 5.8 Comparison between simulated and experimental intrusion curves: (a) POROS I (b) POROS II. Simulation data represent average of 50 realizations and error bars indicate 1 standard deviation from the mean.

••• experimental data — simulation data

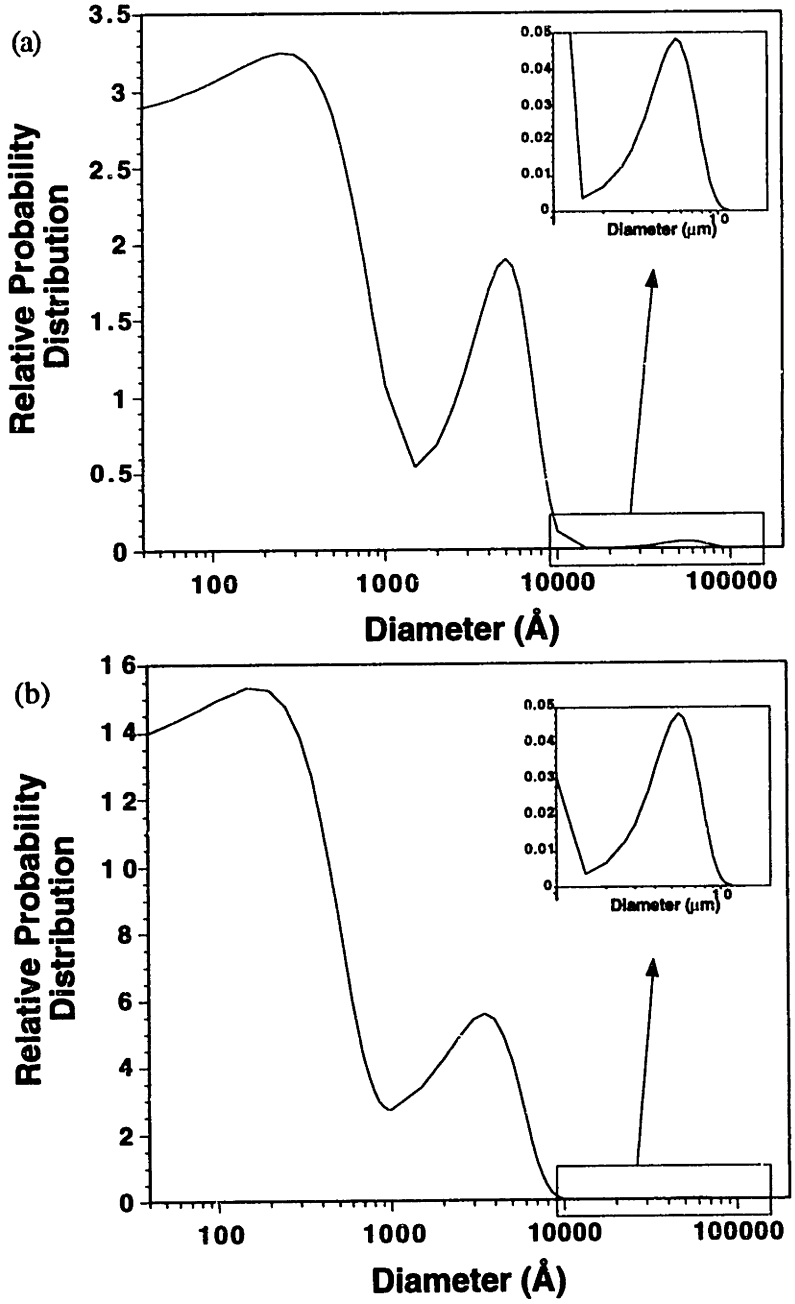


Figure 5.9 Relative probability distribution function obtained from final mercury intrusion simulation: (a) POROS I (b) POROS II.

in pore sizes) where diffusion occurs. The mean macropore diameters of 5000Å and 3500Å obtained from the simulations compare very well with those obtained experimentally based on SEM and image analyses. These results are also consistent with the approximate measurements of the pore sizes inferred from SEM by Afeyan and his co-workers (1990). They have reported macropore sizes of about 6000-8000Å interconnected by diffusive pores of about 500Å. Moreover, a comparison between POROS I and II shows that POROS II has an abundance of smaller pores, in both the macropore and micropore sizes (evident in the relative proportion of micropores to macropores), compared to POROS I, to effect a higher amount of surface area for adsorption. The adsorption capacity has been found to be about 40% more in POROS II compared to POROS I.

Also found from the simulations is the magnitude of the mean pore size of the interstitial pores in the column: a mean size of 5.6μm was obtained. This corresponds to a ratio of pore size to particle diameter of 0.28. In reports of particle size analysis using mercury intrusion measurements, Frevel and Kressley (1963), Mayer and Stowe (1965) and Smith and Stermer (1987) have used different sphere-packing models to elucidate the mean size of the pore spaces between packed spherical particles. They have found that the ratio of the mean size of the pores between these particles to the diameter of the particles ranges from 0.27-0.37 depending on the porosity of the packing. The results inferred using the network model are therefore consistent with the values reported in the literature.

Table 5.2 Summary of Parameters of Pore Size Distribution for POROS I and II

Pore Group	POROS I			POROS II		
	p_i	μ_i	σ_i	p_i	μ_i	σ_i
Macropore	1.0	5000 Å	2100 Å	2.8	3500 Å	2000 Å
Micropore	0.35	250 Å	450 Å	0.98	160 Å	280 Å
Interstitial	0.21	5.6 μm	1.82 μm	0.21	5.6 μm	1.82 μm
PI	2.5			2.5		

The parameters used here are the same as those defined in Equation (4.1) where p_i is the relative mathematical proportion of pores of class i,

μ_i is the mean diameter of pores of class i,

σ_i is the standard deviation of diameter of pores of class i.

One important point to note in the network model used here is the sole use of the mercury intrusion curve to obtain the PSD existing in the particles. Conner and Lane (1984) and Lapidus *et. al.* (1985) have employed throat-pore networks to represent disordered media where the throat size distribution was given by the intrusion curve and the pore size distribution was dictated by the extrusion curve. Using such networks, they were able to explain the existence of hysteresis and mercury entrapment in the mercury porosimetry process on the basis of pore structure. However, there are equally much evidence to show that hysteresis and mercury entrapment are due to the difference in contact angle of mercury during intrusion and extrusion (van Brakel *et. al.*, 1981). In addition, the complex extrusion process can take place in a few different mechanisms which

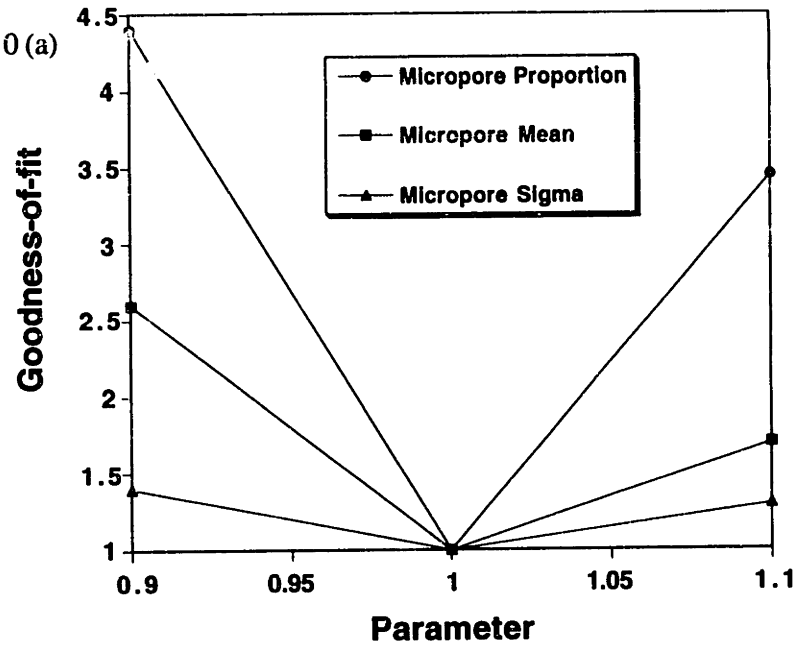
is subject to a great deal of variation (van Brakel *et. al.,* 1981; Modry *et. al.*, 1981 and references therein). A number of different extrusion algorithms are thus possible, yet besides the work of Androutsopoulos and Mann (1979), Tsakiroglou and Payatakes (1990), Park and Ihm (1990) and Portsmouth and Gladden (1991), few of the researchers have described this aspect of their simulations in much detail. Even if full details were available on every algorithm, it would still be difficult to judge the efficiency of these routines. Therefore, rather than having two separate size distributions of the throats and pores defined by the intrusion and extrusion curves respectively, and evaluating each of the many different extrusion mechanisms, the network model presented in this thesis used only the intrusion curve to elucidate the PSD. Moreover, the network model used here may, in many respects, bear a close resemblance to real systems than throat-pore models since there is no structural requirement that each large pore be connected to the rest of the network only by narrow throats. The motivation behind the development of a bond model (instead of a site and bond model as in throat-pore models) is also the desire to produce a network with parameters that could be defined entirely by porosimetry and to use the characterized pore structure in fluid transport studies (Section 5.10).

5.3 Sensitivity Analysis of PSD Parameters and PI

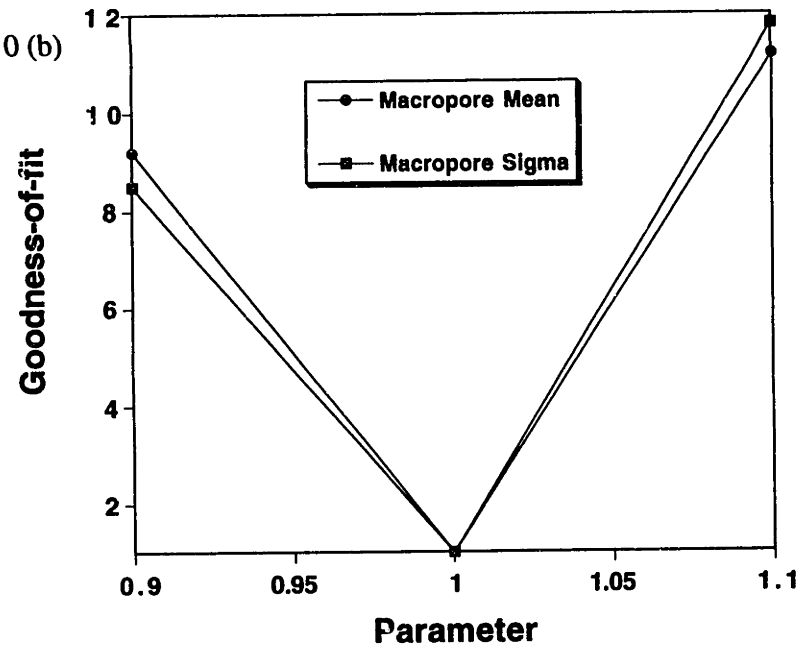
To better understand the nature of the network model and to investigate the uniqueness of the parameters of the PSD and PI obtained for POROS I and II from matching mercury intrusion simulation to experimental data, a sensitivity analysis of the input parameters of the model was performed. For this, the parameters of the PSD and PI were changed such that each of the parameters was first decreased by 10% of the final value of that parameter (hereafter termed the normal value) and then increased by 10% above the corresponding normal value while keeping all the other parameters unchanged. With the new set of input PSD and PI at each analysis, the network lattice was generated and mercury intrusion was simulated. The average goodness-of-fit factor for 50 realizations was then computed. Figure 5.10 shows the results of the sensitivity analysis for input parameters of POROS I while Figure 5.11 shows the results for input parameters of POROS II. In these graphs, the varied parameter and the corresponding goodness-of-fit factor were normalized by the normal values obtained in Section 5.2. The graphs plot the normalized goodness-of-fit factor against the corresponding parameter which was changed in the analysis. As for the parameters of the interstitial pore size distribution and PI, results are only shown for POROS I since the same set of parameters was used for POROS II.

As can be seen from Figures 5.10 and 5.11, decreasing the respective parameter by 10% or increasing the respective parameter by 10% resulted in a fit that has a higher goodness-of-fit factor, that is, a poorer fit. This indicates that the set of input parameters derived from the mercury intrusion simulation produced the best fit, or the minimum average fractional error. In addition, these results also reinforce the uniqueness of the input parameters obtained.

5.10 (a)


Micropore Proportion
Micropore Mean
Micropore Sigma
Goodness-of-fit
Parameter
4.5
4
3.5
3
2.5
2
1.5
1
0.9
0.95
1
1.05
1.1

5.10 (b)


Macropore Mean
Macropore Sigma
Goodness-of-fit
Parameter
12
10
8
6
4
2
0.9
0.95
1
1.05
1.1

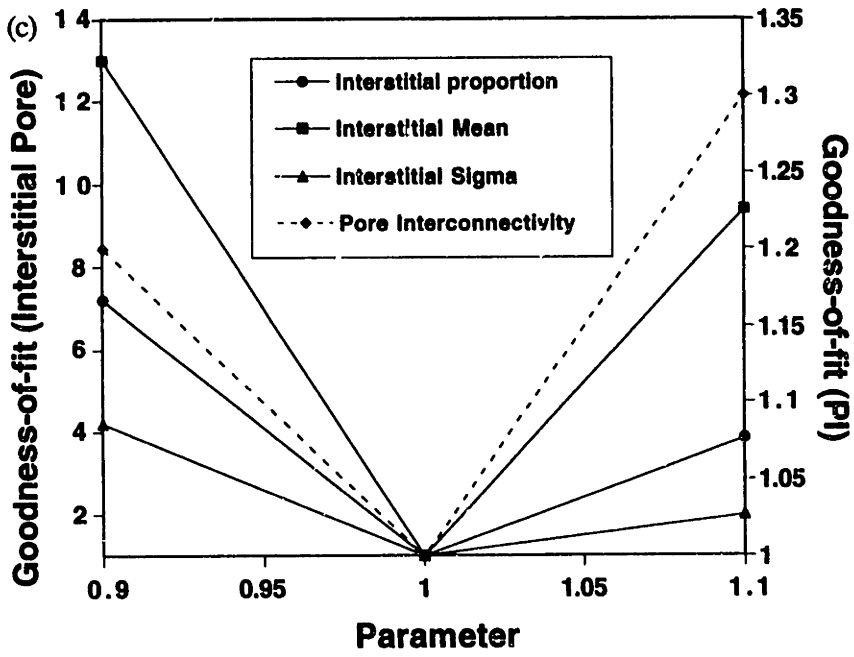


Figure 5.10 Results of sensitivity analysis for changed PSD parameters of POROS I: (a) Parameters of micropore size distribution (b) Parameters of macropore size distribution (c) Parameters of interstitial PSD and PI

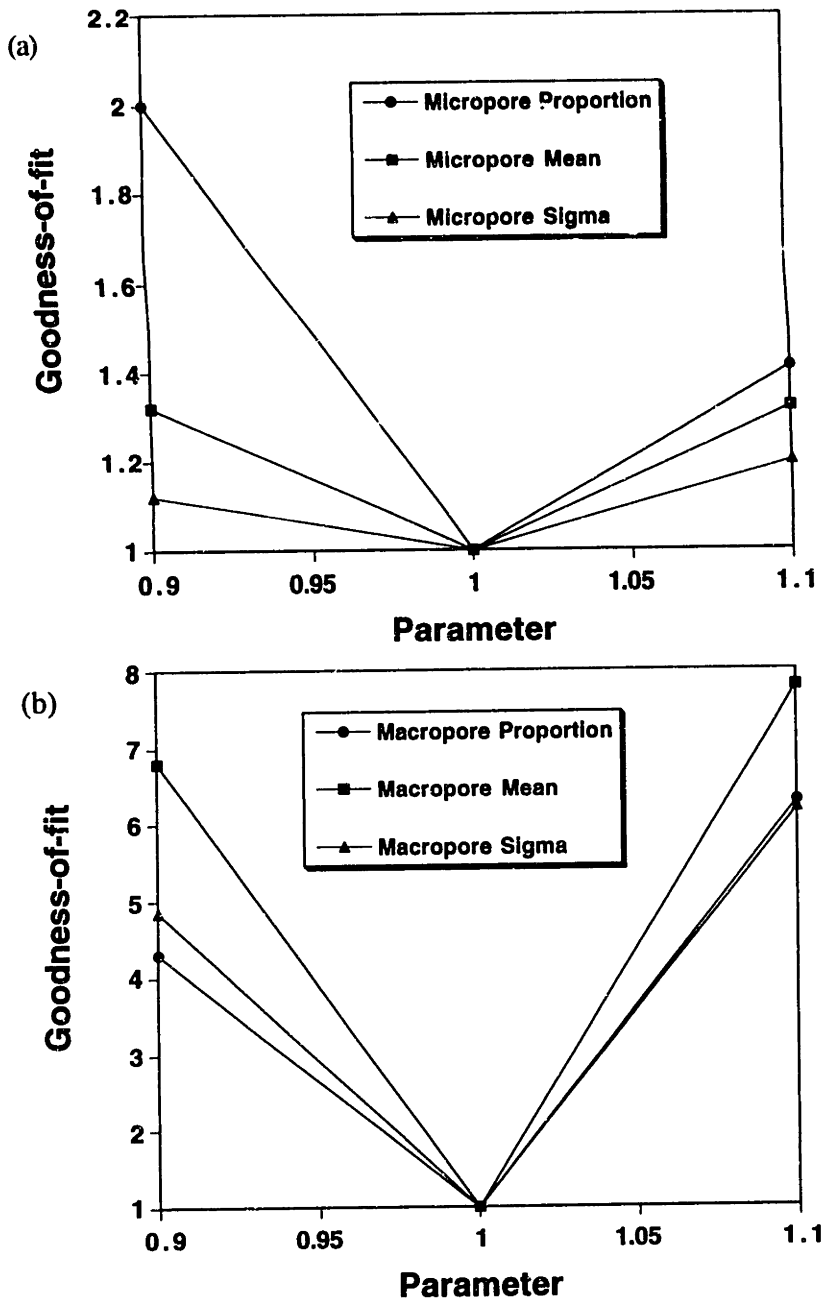


Figure 5.11 Results of sensitivity analysis for changed PSD parameters of POROS II: (a) Parameters of micropore size distribution (b) Parameters of macropore size distribution

Among the parameters that describe the pore size distribution of the perfusive media, the goodness-of-fit was found to be most sensitive to the parameters of the macropore PSD in both cases of POROS I and POROS II. By increasing or decreasing the parameters for the groups of macropores by 10%, the errors in the fit betwee simulation and experimental cumulative intrusion data increased by as little as 400%, but as much as 1200%. This is understandable since in the particles, a great portion of the porosity, of which the cumulative mercury intruded into the sample is indicative, is provided by the macropores. Consequently, a slight change in the sizes or relative numbers of the macropores (evident in changes in the proportion or mean and standard deviation of the macropore PSD respectively) can result in a wide shift of the simulated mercury intrusion curve with a concomitantly bad fit between experimental and simulation data. The same reasoning also holds true for the case of the interstitial pores, where the goodness-of-fit was very sensitive to changes in the parameters of the interstitial pore size distribution.

5.4 Effects of Lattice Size

It is obvious that every computer uses finite computational time and has finite storage memory. While using networks of larger lattice sizes may be more valid for representing an actual porous media, larger networks suffer from having computations which become cumbersome and in some cases, intractable even with the most powerful computers. On the other hand, one has to be careful about how the simulation results are extrapolated to the asymptotic limits of an infinite system. One way to deal with finite size simulations is the use of periodic boundary conditions which have been used in generating the networks in this thesis. Nonetheless, it is necessary to investigate the effects of lattice size on the simulation results. The mercury intrusion simulations for POROS I were performed on cubic networks of LxLxL lattice sizes for L ranging from 10 to 30. For each lattice network, the average goodness-of-fit, F factor, was obtained for 20 realizations and the results are plotted against lattice size, L, as shown in Figure 5.12. Numbers next to the data points represent the average fractional error in each simulation.

Although smaller lattice sizes requires much less computational time and memory, it can be seen from Figure 5.12 that the error involved in the simulations are concomitantly higher. On the other hand, there is not a clear advantage in using large networks (L=30 compared to L=25) in terms of the goodness-of-fit. The CPU time necessary for the L=30 networks compared to the L=25 networks is about 75% more. Therefore, for all the simulation work performed in this research, a lattice size of 25x25x25 was used.

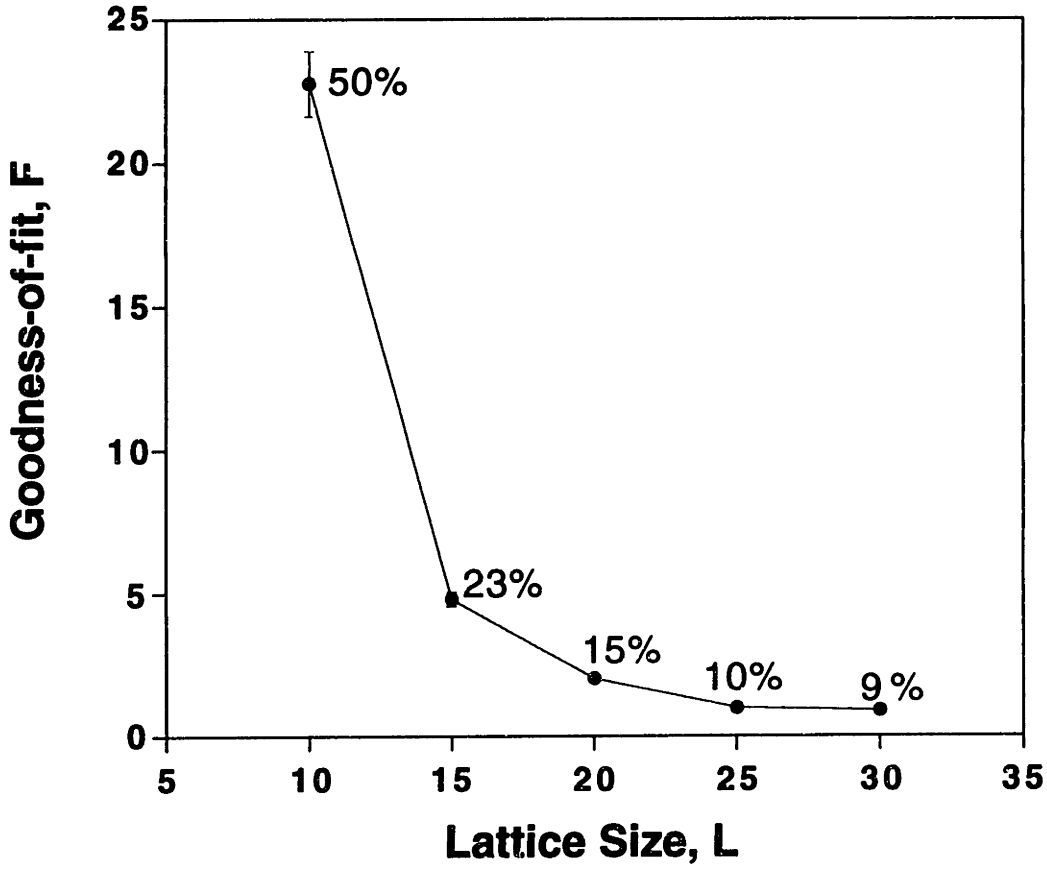


Figure 5.12 Effects of lattice size on goodness-of-fit based on average of 20 realizations. Numbers next to data represents average fractional error from 92 experimental data points.

5.5 Discretization of Pore Number Distribution

In presentation on the PSD so far, the discussion has been limited to the relative parameters of the distributions: namely the mean and standard deviation of the distribution. It will be very useful if the number of pores in the respective groups of pores can be obtained on a per particle (or per unit particle volume) basis. This will not only serve as a good indicator of the pore sizes and relative number abundance of the pores, but it will also facilitate an efficient comparison between different perfusive particles. Since the model used in this research is fundamentally a discrete representation of the porous media, it is possible to discretize the pore number distributions for both POROS I and II. The discretization results are shown in Figure 5.13. In this figure, the number of pores per bead particle is plotted as a function of pore diameter. Similar plots for number of pores per unit bead volume as a function of pore diameter can also be obtained for comparing particles of different sizes. Since the samples used in this study are both norminally 20μm in diameter, the ordinate of number of pores per bead is a sufficient comparison.

As seen from Figure 5.13, the number of pores in POROS II are as much as three-fold greater than that in POROS I for the pore sizes ranging from 20Å to approximately 5000Å. This was intentionally included by the manufacturer to effect more surface area for protein adsorption in the case of POROS II. It must be made known at this point that the porosities for both POROS I and II are similar, at about 60%. At a first glance of the data, there seems to be a contradiction in terms of the porosities of the two media. The apparent question being how is it possible for POROS II to have as much as three-fold more number of pores compared to POROS I and yet maintain the same porosity?

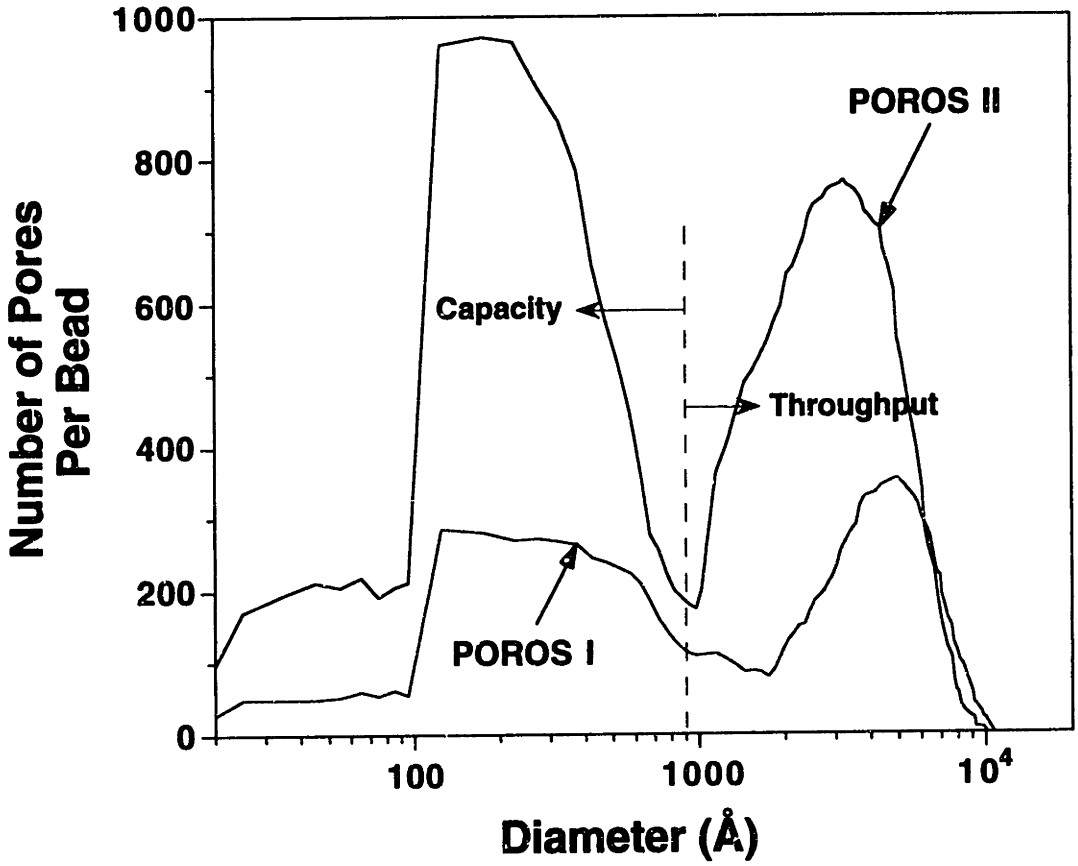


Figure 5.13 Pore number distribution discretized on a per particle basis for POROS I and II.

The answer lies in the pore diameter range between 7000Å and 10,000Å. It is important to recall that the volume of the pores is directly proportional to the cube of their diameters. Therefore, despite the fact that there are three times more pores for POROS II in the smaller and broader diameter range, the two- to three-fold more pores for POROS I in the larger diameter range, albeit narrower, more than make up for the apparent differences in porosities of the particles.

More importantly, these results demonstrate the usefulness of the network model to semi-quantitatively compare the performance (in terms of adsorption capacity and throughput) between the two perfusive packings. The pore number distributions shown in Figure 5.13 can be divided into two parts - one in the pore diameter range from 20Å to 1000Å, and another in the range from 1000Å to about 1μm. The first part represents the diffusive micropores and this assesses the adsorptive capacity of the particles while the second representing the convective macropores assesses the throughput of the particles. Keeping the same particle porosity, the more abundance of micropores in POROS II compared to POROS I provided more adsorptive capacity while maintaining the mechanism of perfusion chromatography (high throughput at low pressure drop) by having sufficient macropores in the particles. This explains the better performace of POROS II compared to POROS I.

5.6 Sensitivity of Statistical Function used to Represent PSD

The results presented in Section 5.1 of SEM and image analysis demonstrate that both the particle size distribution (hence the interstitial PSD) and the macropore size distribution follow a Gaussian distribution for the pore diameters. The type of statistical function to represent the micropore size distribution, however, could not be ascertained due to limitations in the SEM procedure. As such, all of the results presented thus far have been obtained using a Gaussian distribution to describe the micropore size distribution. In this section, a sensitivity analysis compares the use of a lognormal distribution for the micropore size distribution as opposed to using a Gaussian distribution.

The two-parameter log-normal distribution can be represented as:

$$PSD(D) = D\exp\{\frac{-[\ln(D/a)]^2}{2b^2}\} \qquad ...(5.4)$$

where $PSD(D)$ is the pore size distribution function

a and b are parameters of the log-normal distribution

The log-normal distribution was used to represent the micropore size distribution for POROS I and II while the parameters for the macropore size distribution and interstitial PSD obtained for POROS I and II earlier were used in the sensitivity analysis. Mercury intrusion simulation was performed on the resulting network and a match between the experimental and simulated mercury intrusion data was made to infer the parameters for the log-normal micropore size distribution. A total of 50 realizations were performed and the parameters were obtained when the goodness-of-fit was less than 1.0, the

same as that obtained in the simulations using Gaussian distributions to describe the micropore size distribution. The final parameters obtained for the log-normal distribution are shown in Table 5.3.

Table 5.3 Parameters of the Log-Normal Distribution used to Represent the Micropore Size Distribution

Packing	Parameter a	Parameter b
POROS I	130Å	0.80
POROS II	110Å	0.71

To illustrate how these parameters relate to the mean and standard deviation of the micropore size distribution, a plot of the discretized pore number distribution as a function of pore diameter was made. Figure 5.14 compares the discretized pore number distribution (on a per particle basis) for a Gaussian distribution with a log-normal distribution for the micropores for (a) POROS I and (b) POROS II. The data reveal that the pore number distribution obtained, at the best fit between the experimental and simulated intrusion data, using log-normal distribution and Gaussian distribution for the micropores are comparable. Both distributions have approximately the same mean micropore diameter of about 250Å for POROS I and 160Å for POROS II. From these results, it can be concluded that the mercury intrusion simulation is not very sensitive to the type of statistical function used to represent the micropores as long as the same overall goodness-of-fit between the experimental and simulated data is maintained. As such, the results presented earlier are valid and subsequently, the Gaussian distribution was used to represent the

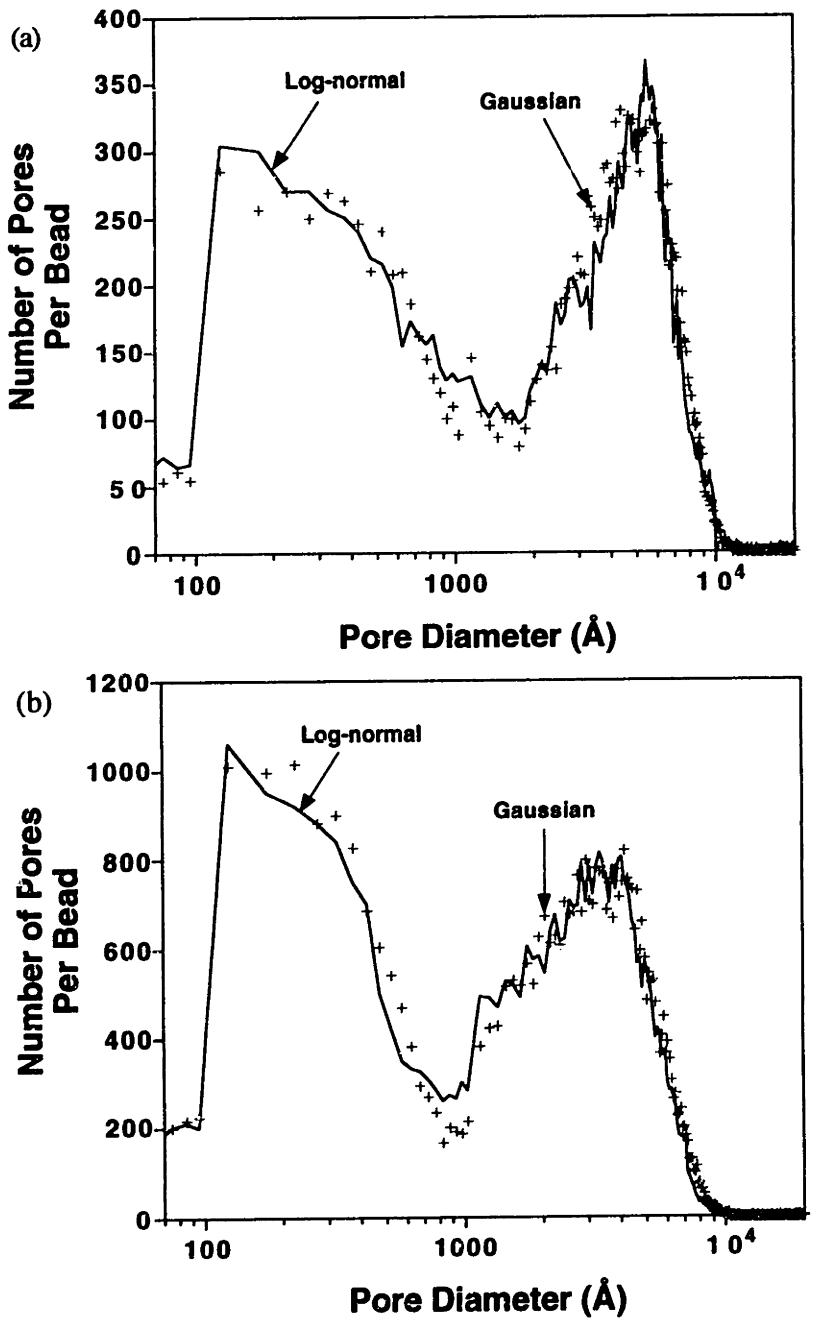


Figure 5.14 Comparison of the discretized pore number distribution for Log-normal distribution and Gaussian distribution for the micropores in (a) POROS I and (b) POROS II

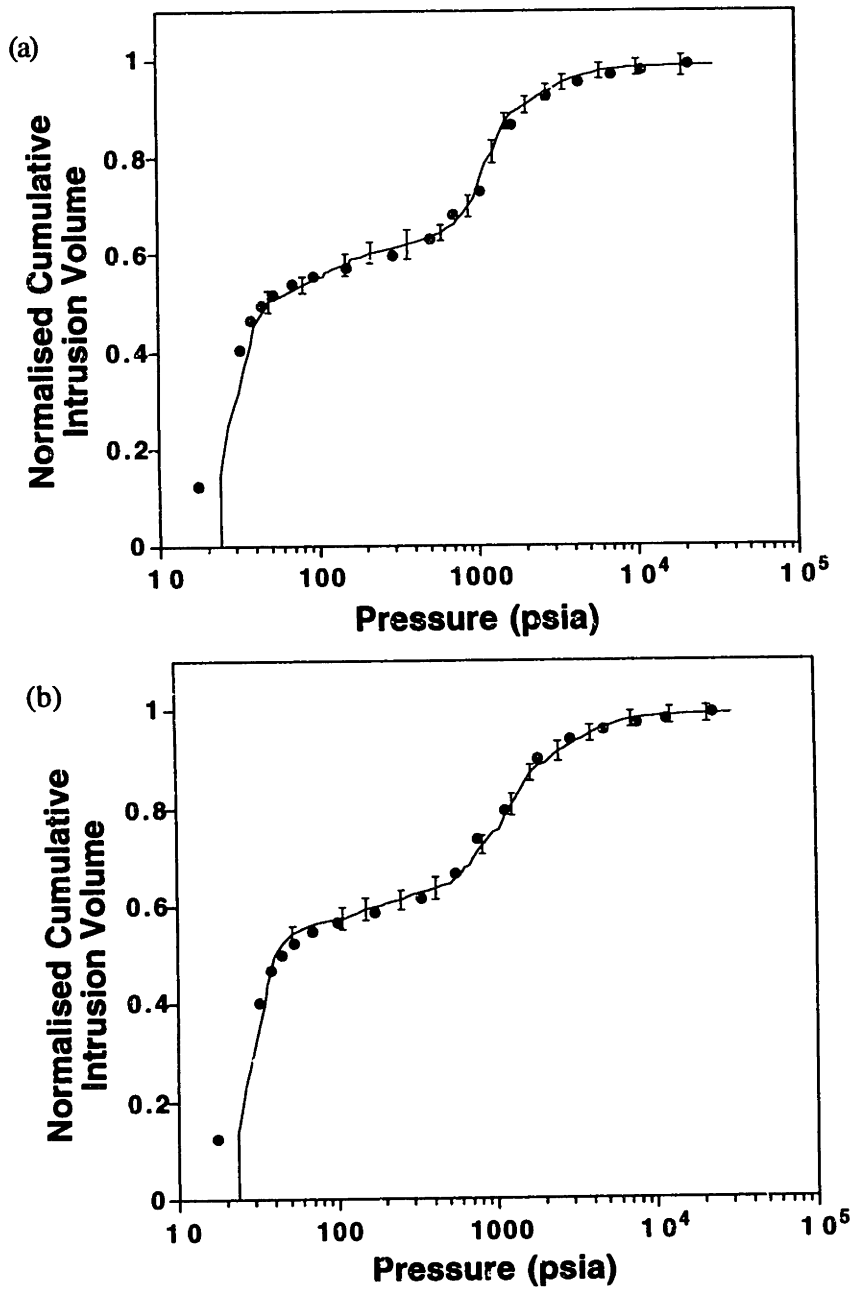
micropore size distribution for consistency with the use of Gaussian distributions for the macropores and the interstitial pores.

In what follows, data will be presented to show that experimental data pertaining to the surface area and porosity of the particles can be elucidated from the network model without the use of adjustable parameters.

5.7 Mercury Intrusion of Blends of POROS I and II

To demonstrate the use of the network model to predict experimental results based on mercury porosimetry, samples of POROS I and II were blended in various proportions (25%, 50% and 75% of POROS I in the blends) and mercury intrusion experiments were performed on the resulting samples. Using the input parameters independently obtained for POROS I and POROS II, lattice networks were generated for the blends and simulations of mercury intrusion into the networks were carried out. Simulation and experimental results are compared in Figure 5.15 for the different blends. Again, 50 realizations were performed for each of the simulation experiment. With no adjustable parameters, it was possible to obtain an excellent agreement between the experimental and the simulation data for all the three blends. The goodness-of-fit ranged from .32 to 1.80, corresponding to average fractional errors of only 12 to 14%.

Figure 5.16 shows a replot of the simulation data for the 25/75 POROS I/II blend and the 75/25 POROS I/II blend to illustrate the sensitivity of the network model in simulating the mercury porosimetry experiments for the various blends. The uniqueness of the PSD parameters of the blends was investigated by elucidating independently the parameters of the PSD for each of the blends. In this study, each blend was treated as an unknown sample in which its PSD had to be elucidated. Using the mercury intrusion simulation technique, parameters of the PSD for each of the blends were "backed-out" such that the goodness-of-fit between the experimental and the simulated intrusion curves was less than 1.00. A summary of the parameters for each of the three blends are shown in Table 5.4.


(a)
Normalised Cumulative Intrusion Volume
0
0.2
0.4
0.6
0.8
1
10
100
1000
10^4
10^5
Pressure (psia)
(b)
Normalised Cumulative Intrusion Volume
0
0.2
0.4
0.6
0.8
1
10
100
1000
10^4
10^5
Pressure (psia)

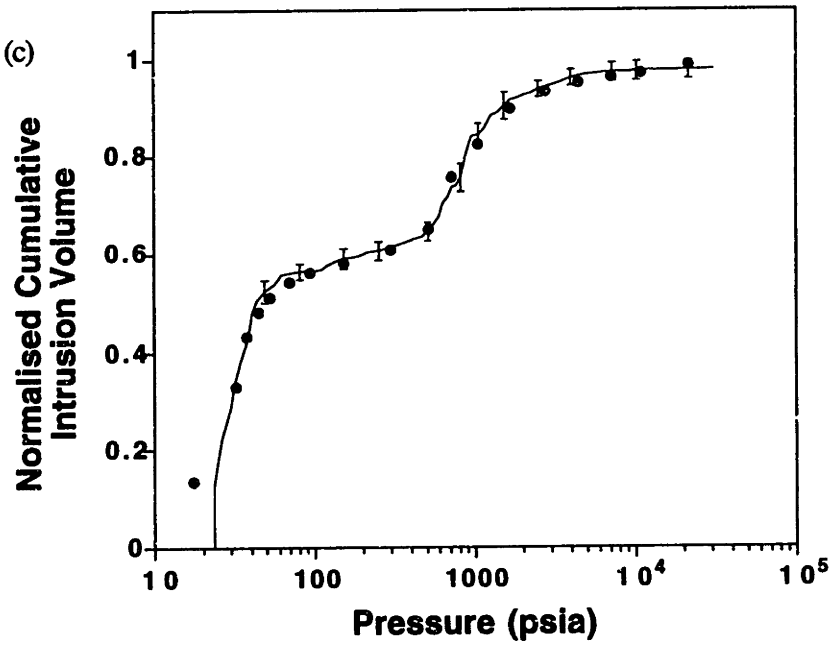


Figure 5.15 Comparison between simulated and experimental intrusion results of blends of POROS I and II:

(a) 25/75% POROS I/II; (b) 50/50% POROS I/II; (c) 75/25% POROS I/II

Simulation data represent average of 50 realizations and error bars indicate 1 standard deviation from the mean

••• experimental data — simulation data

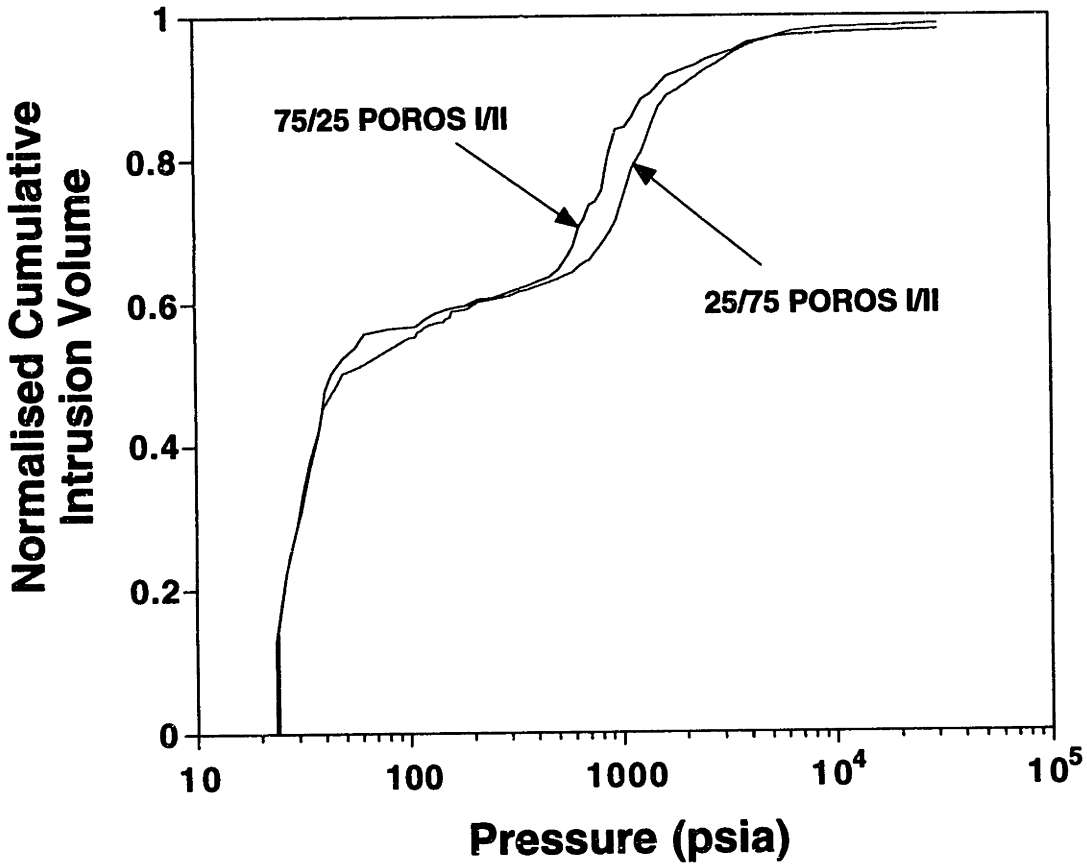
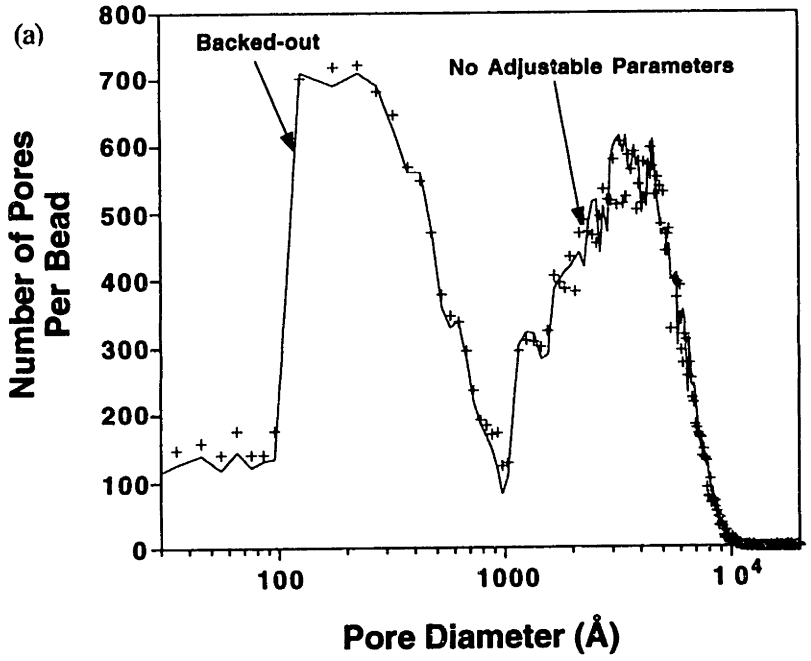


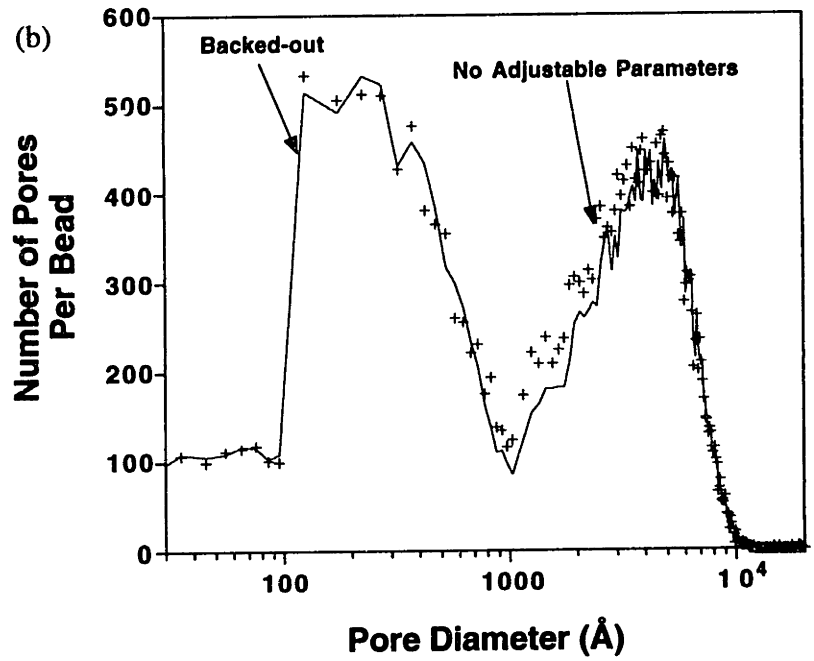
Figure 5.16 Comparison of simulated intrusion curve for the 25/75% POROS I/II blend and the 75/25% POROS I/II blend to demonstrate the sensitivity of the network model.

Table 5.4 Parameters of PSD for Blends of POROS I and II obtained from matching Intrusion Curves

Blend (POROS I/II)	Macropore Size Distribution		Micropore Size Distribution	
	Mean Diameter (Å)	Standard Deviation (Å)	Mean Diameter (Å)	Standard Deviation (Å)
25/25	4200	2100	170	350
50/50	4400	2100	180	350
75/25	4600	2000	200	370

Figure 5.17 plots the discretized pore number distribution as a function of pore diameter for the three blends. In each graph, the pore number distribution computed from parameters which have been "backed-out" from the match between experimental and simulation intrusion data is compared with that which was computed from the parameters of PSD of POROS I and II which have been inferred earlier. In the latter case, no adjustable parameters were used to match the experimental intrusion curves. The results in each panel of Figure 5.17 reveal that the same PSD was obtained in both cases for all the three blends. In addition to validating the network model for simulating mercury porosimetry experiments, the results in this section further reinforce the conclusion that the PSD parameters obtained for POROS I and II are indeed unique.


(a)
Backed-out
No Adjustable Parameters
Number of Pores Per Bead
800
700
600
500
400
300
200
100
0
100
1000
10^4
Pore Diameter (Å)


(b)
Backed-out
No Adjustable Parameters
Number of Pores Per Bead
600
500
400
300
200
100
0
100
1000
10^4
Pore Diameter (Å)

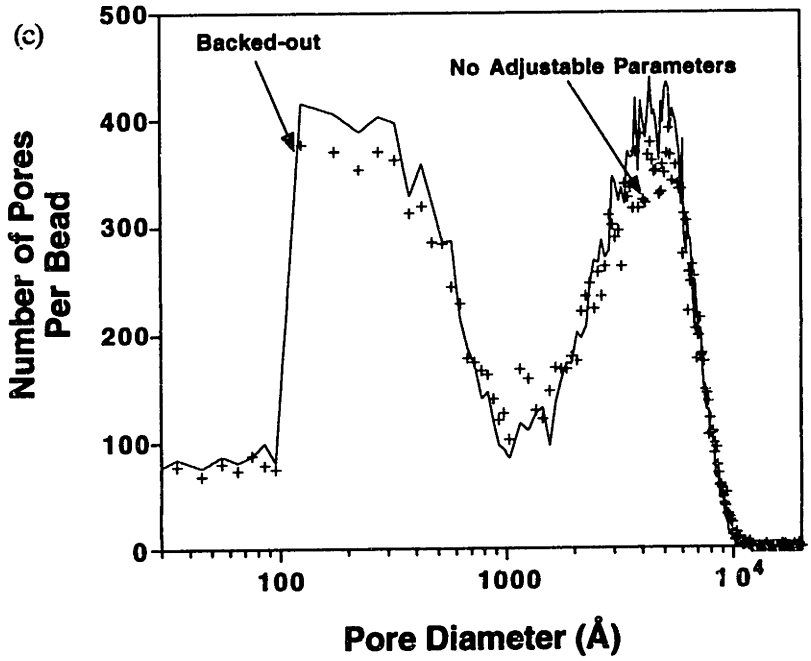


Figure 5.17 Comparison between the discretized pore number distribution computed from parameters "backed-out" from simulation match with experimental data (solid line) and that computed from PSD parameters of POROS I and II without adjustable parameters (+++)

(a) 25/75% POROS I/II (b) 50/50% POROS I/II (c) 75/25% POROS I/II

The important implication of this study is the possibility of using the model for quality control and quality assurance applications. For a random batch of the packing material from the production line, it is now possible to determine if the porous structure manufactured are within specifications by not just simulating the pure sample intrusion but also blends of the random batch with known, characterized batches and comparing the simulated data with experimentally obtained data.

5.8 Protein Adsorption Capacity Studies

In an effort to investigate the application of the model to elucidate surface accessibility of the media, experiments on equilibrium adsorption capacity studies were performed on POROS I and II for proteins of various molecular weights and hence molecular sizes. Figure 5.18 shows the adsorption isotherms of POROS I and II for each of the five proteins investigated. These adsorption isotherms were fitted with the Langmuir-Hinshelwood kinetics to derive the maximum static capacities and the affinity constants for the various proteins on POROS I and II. The results are tabulated in Table 5.5. The standard errors involved in the estimation of the maximum static capacity ranged from 0.02% to 1.2% for a 95% confidence interval while the error in estimating the affinity constant ranged from 0.01% to 2.5% for the 95% confidence interval. The higher errors were incurred for the cases of lysozyme and ovalbumin. This resulted from the use of protein concentrations which were too low to characterize the adsorption isotherm. Nevertheless, since the errors were less than 1.2% in estimating the maximum static capacity, the errors are not significant.

Figure 5.19 plots the absolute protein capacities for the two packings. For all of the proteins studied, it can be seen that POROS II provide more accessible area for adsorption compared to POROS I. In comparing the accessibility of adsorption surface area as a function of the protein size, however, the ratio of adsorption capacity of POROS II to I is more meaningful than the absolute static capacity. In using the ratio, surface chemistry between the proteins and the adsorptive surface can be eliminated and protein

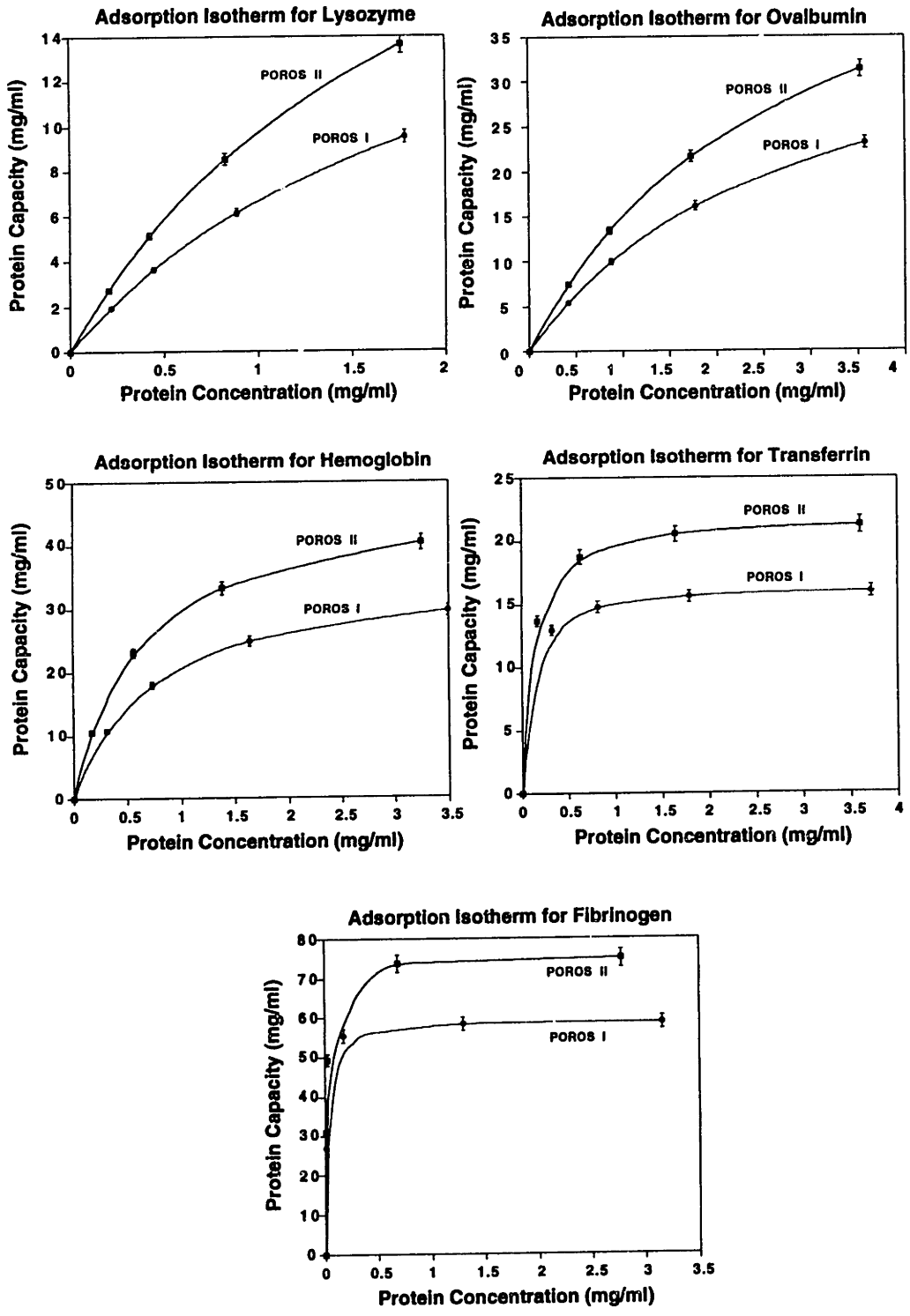


Figure 5.18 Adsorption isotherms of POROS I and POROS II for the different proteins tested.

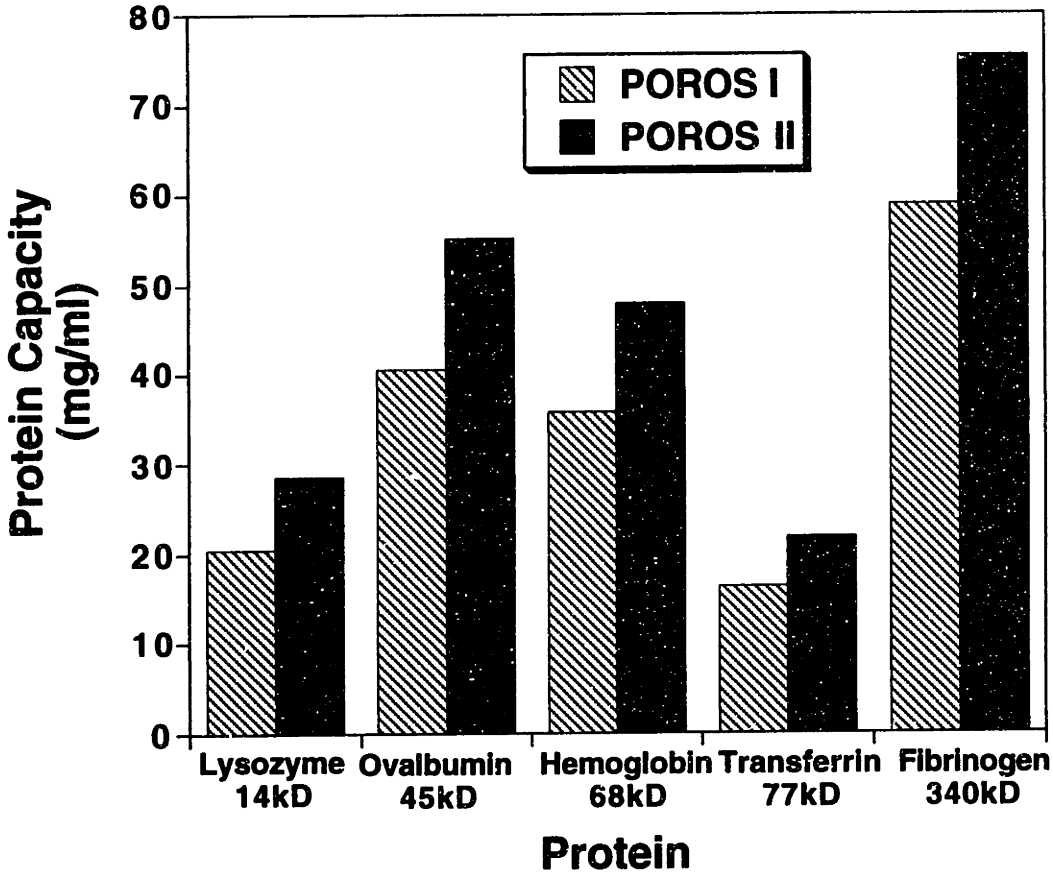


Figure 5.19 Equilibrium protein adsorption capacity of POROS I and POROS II with proteins of different molecular weights.

Table 5.5 Parameters of Langmuir-Hinshelwood Adsorption Isotherm

Protein	POROS I		POROS II		$q_{max}II/q_{max}I$
	Kp	qmax	Kp	qmax	
Lysozyme	2.065	20.54	1.924	28.55	1.39
Ovalbumin	2.709	40.57	2.695	55.18	1.36
Hemoglobin	0.721	35.80	0.606	47.97	1.34
Transferrin	0.086	16.40	0.101	21.81	1.33
Fibrinogen	0.011	58.89	0.013	75.38	1.28

accessibility can be evaluated on a wholly physical basis. This will be possible if the particles under comparison have the same surface chemistry with respect to the proteins under examination. From Table 5.5, it can be seen that the Kp values obtained for POROS I and II are comparable with a maximum difference of only 15%. Therefore, it can be concluded that POROS I and II both have the same surface chemistry for each of the proteins investigated and the assumption of equating the ratio of the protein capacities to the ratio of the accessible surface areas of the particles is justified.

Using the solid sphere model, accessible surface areas of the packings were simulated as a function of protein size. A plot of experimental and simulated results is shown in Figure 5.20. For comparison, the simulated data for the bundle of parallel tube model are also shown. From Figure 5.20(a), it can be seen that the match between experimental and simulation data (network

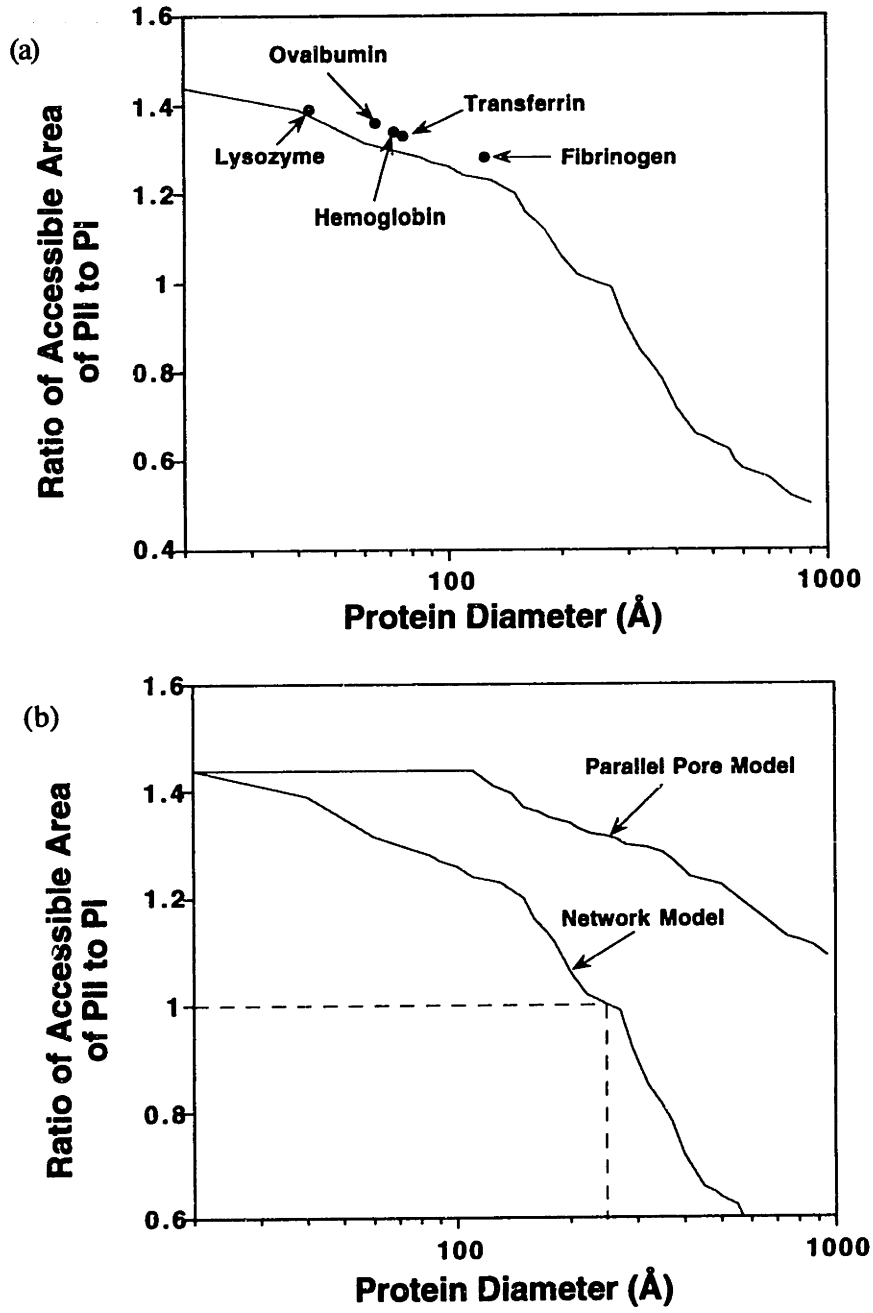


Figure 5.20 Ratio of accessible areas for POROS II to POROS I for different protein sizes.

(a) Comparison between experimental and simulation data;

•○• experimental data; — simulation data;

(b) Comparison between network model and parallel pore model

model) is excellent, providing a corroboration for the use of the network model in predicting surface accessibility of proteins. More importantly, these results demonstrate the important role that the pore interconnectivity plays in surface area accessibility in the porous structure of the media (Figure 5.20(b)). With a parallel pore model where there is no interconnectivity, solute molecules can access the pores in the particle uninterrupted and the ratio of the accessible areas remains almost constant up to a solute diameter of about 100Å.

The other important observation from the simulation data is that with a solute diameter of about 250Å, there does not seem to be a competitive advantage of POROS I over POROS II in terms of extra surface area for protein adsorption. This is because with such a large solute, a significant portion of the surface area in POROS II cannot be accessed due to the shielding of the pores which contribute to surface area for adsorption by other smaller pores in the interconnected porous medium. Due to the inability to find large enough protein molecules in the 100's Å diameter range, the second half of the curve (Figure 5.20(a)) could not be verified. Notwithstanding this, the simulations shown in this section provide a good independent validation of the established network model. In the next section, the model will be validated with size exclusion chromatography (SEC) data where solute molecules of large diameters can be used.

5.9 Size Exclusion Chromatography

Apart from being able to predict experimental data based on surface area accessibility, the network model can also elucidate experimental data based on volume accessibility. For this, size exclusion chromatography was simulated using the model, and the data compared with the experimental results. The simulation technique, based on the partition coefficient given by Equation (4.5), had been outlined in Section 4.6. Parameters of the polystyrene standards used in the size exclusion chromatography experiments are summarized in Table 5.6.

Table 5.6 Relevant Parameters of Polystyrene Standards Required for Computation in Equation (4.5)

Molecular Weight (kD)	n=Number of segments of macromolecule	rg(Å)=Radius of gyration	l=Segment length of macromolecule
7.5	70	27	7.8
25	240	55	8.7
48	460	80	9.2
90	870	116	9.7
207	1990	190	10.4
900	8650	450	11.9
3000	28800	913	13.2

Figure 5.21 shows a comparison between simulated and experimental data for the partition coefficient, K, as a function of the gyration diameter of the polystyrene molecules for POROS I and II. The agreement between the simulated and experimental data are excellent. For POROS I, errors between the two sets of data varied from 4% to 10%, while for POROS II, they ranged from 5% to 7%. These results reinforce the original derivation of Giddings and his co-workers (1968) that the partitioning of molecules between bulk fluid and inert, porous networks is based on a steric interaction between the solute and the pore.

Furthermore, these results demonstrate the inconsistencies in earlier SEC work by Halasz and Martin (1978) who had assumed that the size exclusion of any molecule is based on total exclusion regardless of the relative size of the molecule to the pore diameter. In their analysis, any molecule of radius less than the radius of the pore will have an exclusion coefficient of one. That assumption had forced them to conclude that the polystyrene molecules used in their experimental work had an apparent radius two and a half times greater than that evaluated from the Van Krefeld and Van den Hoed (1973) formula (Equation (3.4)). More importantly, the results reported here provide another independent corroboration for the network model that has been established, and a relevant confirmation for the parameters of the PSD and PI obtained for POROS I and II. Furthermore, the excellent agreement of the data indicate that the discrete network model can offer a useful approach for the specification and design of column media for SEC applications.

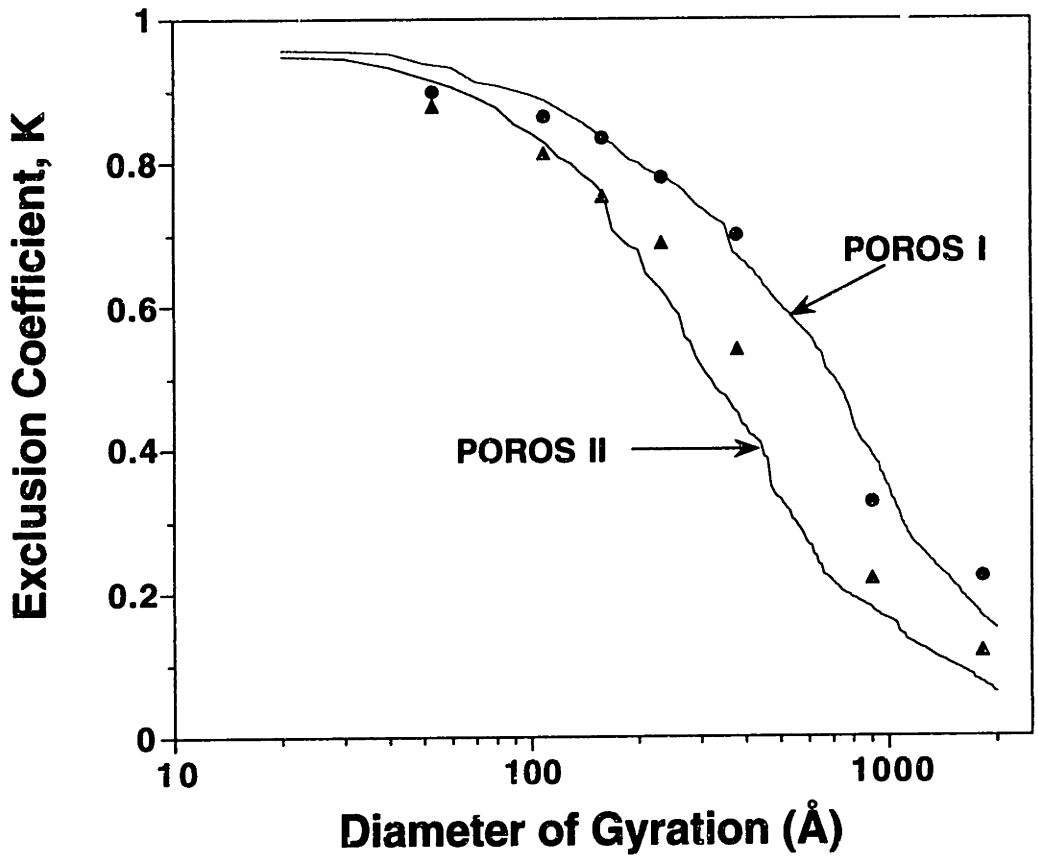


Figure 5.21 Size exclusion chromatography for POROS I and POROS II.

• POROS I, experimental data; ▲ POROS II, experimental data

5.10 Intraparticle Convection and Column Efficiency

The amount of bandspreading in an HPLC column has traditionally been used as an indication of the efficiency of the column. By measuring the spread of the elution peak of a non-retained solute through the packing, the column efficiency is quantified by means of the HETP or height equivalent to a theoretical plate. This can be made dimensionless upon division by the particle diameter to obtain the reduced plate height. Theoretical analyses of the effects of intraparticle convection on column efficiency have been developed by several authors (see Section 2.3.2). All the analyses have been based on macroscopic models in which pore structure and pore size distributions have been lumped into packing porosity and the empirical parameter of tortuosity. In this section, results will be presented on the use of the network model developed to quantify intraparticle convective flow subject to changes in pore size distribution (PSD) and pore interconnectivity (PI). First and foremost, it seems apt to provide a discussion of the efficiency of a column packed with perfusive media.

The residence time distribution (RTD) of lysozyme under non-retained conditions was obtained experimentally on packed columns of POROS I, POROS II and blends of the two media. The reduced plate heights were calculated based on the amount of bandspreading of the eluite at superficial velocities ranging from 70cm/h to 4000 cm/h. Figure 5.22 plots the van Deemter curves (reduced plate height as a function of superficial velocity) for the different perfusive columns. The blends of POROS I and POROS II were used to illustrate the effects of different PSD on the efficiency of the perfusive column. Since POROS I and POROS II have different PSD, a column packed with

blends of the two media will naturally have different overall PSD's compared to either POROS I or POROS II. The results in Figure 5.22 show that as a larger proportion of POROS II is packed into the column (indicating a shift in the PSD to smaller mean macropore and micropore sizes), the efficiency of the column decreases (evident by an increase in HETP). The PSD of the packing can therefore be adjusted to effect a change in the efficiency of the column. However, a tradeoff is quite evident. It is apparent that the efficiency of the column can be improved by using packings of large pore sizes, but it is important to note that increasing the sizes of the pores will correspondingly reduce the effective adsorption capacity caused by a reduction in the available surface area.

A point noteworthy here is the independence of the column efficiency on the porosity of the packings. As noted earlier, both POROS I and POROS II have nearly identical porosities and the columns used in the experiments all have the same total porosity (ε_b=0.70, obtained from acetone spike-experiments). These results illustrate the fact that particle porosity is not the only determining factor in assessing different packings in columns: contrary to common usage in modelling work using continuum macroscopic models (Szekely and Propster, 1975; Rodrigues *et. al.*, 1991). The parameters, PSD and PI of the packings, are also important determining factors for the differences in intraparticle convective flow in the packings. It will be demonstrated in the later sections the application of the network model to elucidate these differences for various perfusive media.

The experimental data also show that in perfusion chromatography , the reduced plate height reaches a plateau at increasing superficial velocity for all

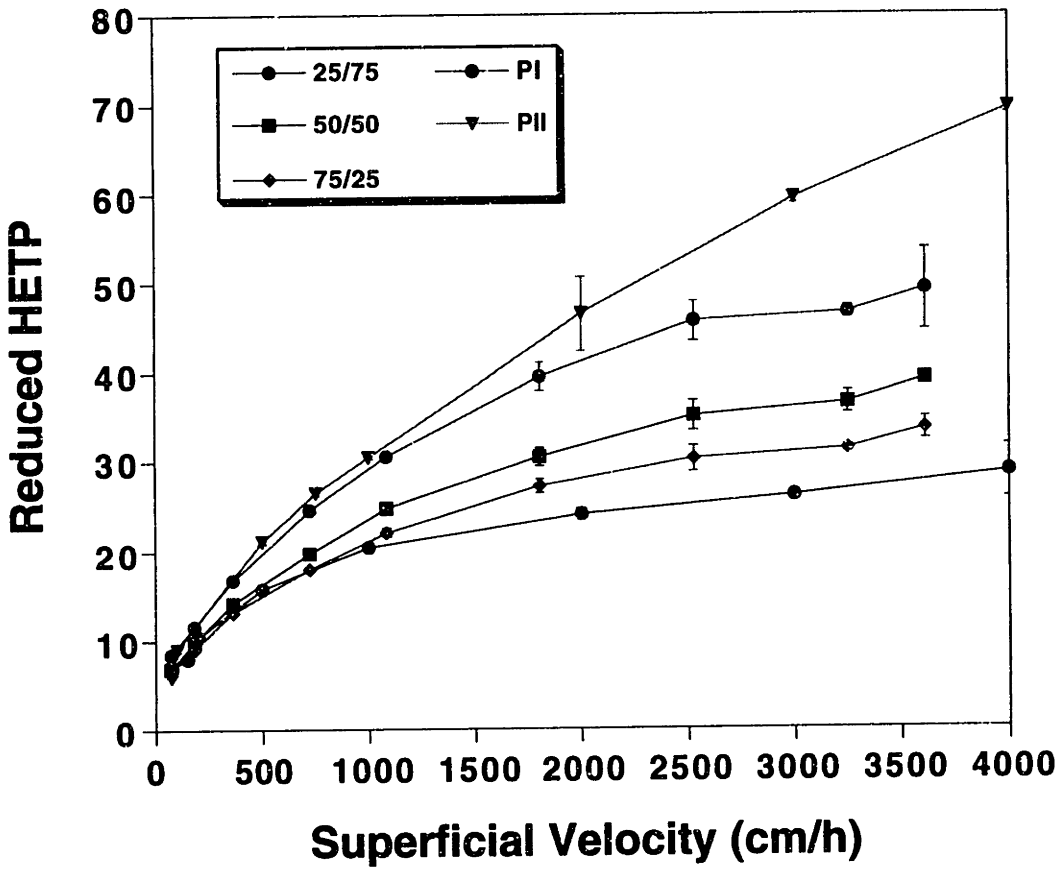


Figure 5.22 Reduced plate height (column efficiency) as a function of superficial velocity. Experimental conditions of non-retained lysozyme were: acetonitrile (ACN)-water (70:30), 0.1% trifluoroacetic acid (TFA), 1mg/ml, 1.0μl injection, dectection at 215nm. Columns were 4.6mm x 100mm. PI: POROS I; PII: POROS II; i/j: POROS I and II blended in mass ratios of i to j.

the five columns tested. The implication of this being that perfusion chromatography can be operated to high flow rates without sacrificing on column efficiency as opposed to conventional HPLC packings where plate heights increase monotonically with linear velocity (van Deemter, 1956; Afeyan *et. al.,* 1990).

An interesting observation of the results shown in Figure 5.22 is the convergence of the plate heights at low superficial velocities. At superficial velocities less than about 500 cm/hr, the column efficiencies for the five columns are similar and independent of the size of the convective macropores in the media. This is because at the lower flow rates, the value of the intraparticle Peclet (Pe) number becomes so small that the influence of intraparticle convection is unimportant. As a result, perfusive packings become advantageous when high throughput separations are required and when the number of theoretical plates necessary to bring about the separation is moderate (Frey *et. al.,* 1993). Additionally, as the column efficiency decreases (from POROS II to POROS I), the superficial velocity at which a plateau of the plate heights is reached became higher. For the packing with the lower column efficiency, a higher superficial velocity is necessary to trigger intraparticle convection (Pe=1) due to a lower intraparticle velocity in such a medium. Therefore, in optimizing the use of perfusive packings, the minimum superficial velocity for the onset of intraparticle Pe=1, where intraparticle convection becomes influential, is an important consideration apart from adsorption capacity. In the final sections of this chapter, discussion on optimizing the PSD of the perfusive media with regards to adsorption capacity and intraparticle convection will be presented. The next section will present results obtained from

applying the transport methodology described in Section 4.7 on the network model to elucidate intraparticle convective flow.

5.10.1 Medium Permeability Using Network Model

A fixed flow rate of 1000 cm/h was imposed on the network and the background pressures at the nodes were computed. In theory, any flow rate can be used and the pressure drop of the network will be scaled accordingly. A total of 75 realizations were performed to reduce the standard deviation of the pressure drop to less than 5% of the average value. Figure 5.23 shows the match between simulated and experimental pressure drops. For both POROS I and II, the agreement between the experimental data and the simulation was good; although the network model predicted a slightly higher pressure drop (average of 11% higher in POROS II and 9% higher in POROS I) in both cases. Based on the background pressures computed at the lattice nodes, the flow rate through the interstitial pore cluster was computed and the corresponding flow rate in the intraparticle pores (through the particle) was calculated by subtraction. Table 5.7 tabulates results of the flow rate through the particle as a percentage of the total flow through the bed for POROS I, II and blends of POROS I and II. Also shown in Table 5.7 are the corresponding velocity ratios, given by Equation (4.11), reiterated here:

$$K_u = \frac{u_p}{u_b} = \frac{Q_p}{Q_b} \frac{1}{(1-\varepsilon_b)} \qquad ...(5.5)$$

As the size of the macropore increases from 3500Å in POROS II to 5000Å in POROS I, the percentage of intraparticle flow increase from 0.23% to 0.60% and the corresponding intraparticle convective velocity as a fraction of superficial bed velocity increases from 2.99×10^{-3} to 8.00×10^{-3}. From these

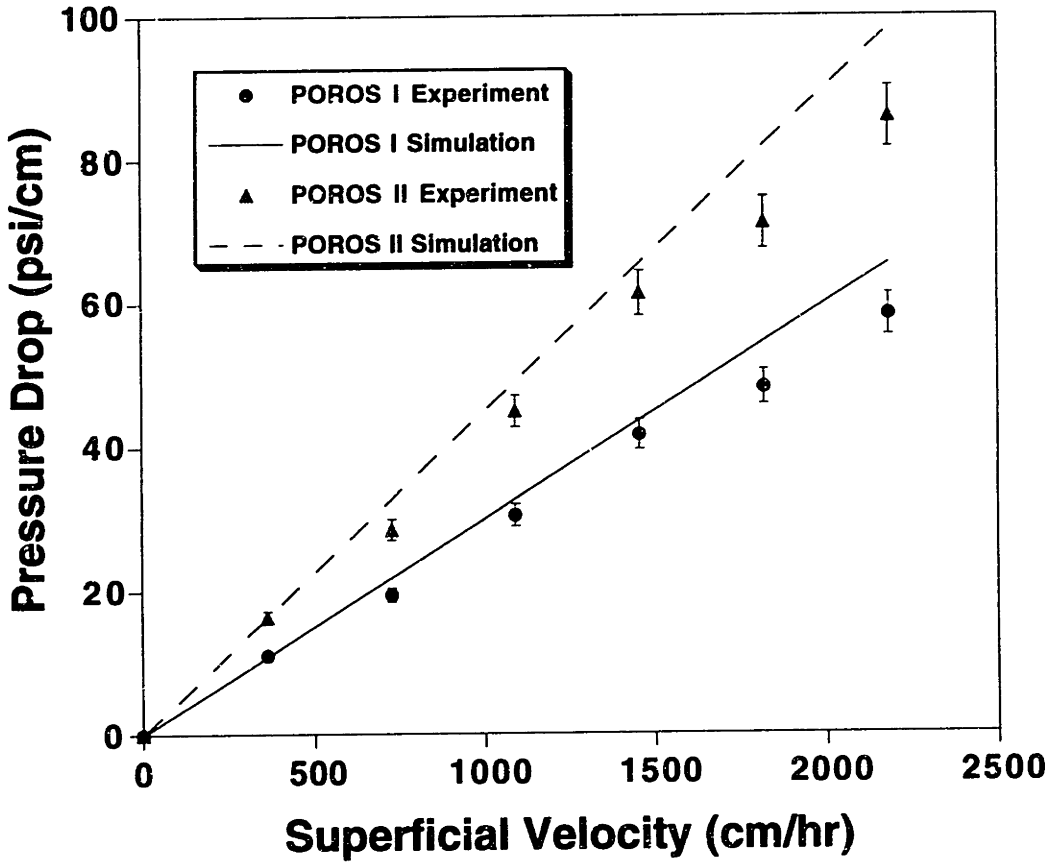


Figure 5.23 Comparison between simulated pressure drop and experimental pressure drop for POROS I and POROS II. 50 realizations were performed for the computer simulations so that one standard deviation from the average pressure drop at each velocity is less than 5%.

results, it can be seen that the amount of intraparticle convection required to invoke perfusion chromatography is indeed minimal. As little as only 0.2% of the flow rate through the particles is sufficient to augment the molecular diffusivity such that convection-controlled transport is triggered in the packings. In order to validate these results, the next two sections will compare the results obtained from existing macroscopic modelling analyses.

Table 5.7 Intraparticle Convective Flow as Computed from Network Model

Medium	Q_p/Q_b (% Through-flow)	$K_u \times 10^3$ (Ratio of Intraparticle Velocity to Superficial Velocity)
POROS I	0.60	8.00
75/25 POROSI/II	0.41	5.54
50/50 POROSI/II	0.38	5.07
25/75 POROS I/II	0.31	4.08
POROS II	0.23	2.99

5.10.2 Medium Permeability Using A Modified Blake-Kozeny Approach

Park and Stephanopoulos (1992) used a modification of the Blake-Kozeny equation to describe and experimentally demonstrate the existence and magnitude of intraparticle convection in ceramic particles used for the immobilization of mammalian cells. In their analysis, the Blake-Kozeny equation

used to describe permeability in a packed bed of non-porous particles was supplemented with a term similar to the Blake-Kozeny term to account for intraparticle convection in the ceramic medium. The relevant equations are simplified and presented as follows:

The total liquid flow rate through the bed, Q_{total} is apportioned between the total flow which passes through the interstitial pores, Q_{inst}, and that which flows through the pores of the beads, Q_{pore} :

$$Q_{total} = Q_{pore} + Q_{inst} \qquad ...(5.6)$$

Using the Blake-Kozeny equation, Q_{inst} and Q_{pore} can be written as:

$$Q_{inst} = S\frac{\varepsilon_b^3}{150(1-\varepsilon_b)^2}\frac{\Delta P}{2\mu L}D_p^2 \qquad ...(5.7)$$

$$Q_{pore} = S(1-\varepsilon_b)\varepsilon_p\frac{\Delta P}{2\mu L}R_{h,pore}^2 \qquad ...(5.8)$$

where S is the cross-sectional area of the packed bed,

ε_b is the bed porosity,

ε_p is the bead porosity,

$\frac{\Delta P}{L}$ is the pressure drop across the bed,

μ is the viscosity of the liquid,

D_p is the bead diameter,

$R_{h,pore}$ is the hydraulic radius of the pores in the beads.

Therefore,
$$\frac{Q_{total}/S}{(1-\varepsilon_b)\frac{\Delta P}{2\mu L}} = \varepsilon_p R_{h,pore}^2 + \frac{1}{150}\left(\frac{\varepsilon_b}{1-\varepsilon_b}\right)^3 D_p^2 \qquad ...(5.9)$$

For particles having the same porosity and pore structure (hence $R_{h,pore}$), Equation (5.9) can be rewritten as :

$$K_{p-s} = \frac{u_b}{(1-\varepsilon_b)\frac{\Delta P}{2\mu L}} = A + BD_p^2 \qquad ...(5.10)$$

where A and B are constants, u_b is the superficial velocity through the bed and K_{P-S} is the total permeability through the packed bed as defined by Park and Stephanopoulos. Based on this analysis,

$$\frac{Q_p}{Q_b} = \frac{Q_{inst}}{Q_{total}} = \frac{A}{A + BD_p^2}$$

that is,
$$\frac{Q_p}{Q_b} = \frac{1}{1 + \frac{B}{A}D_p^2} \qquad ...(5.11)$$

By measuring the pressure drop through the packed columns at various superficial velocities using perfusive medium of different particle sizes, a plot of K_{p-s} against D_p can be made. As a result, the constants A and B can be obtained as the intercept and slope of the linear plot, and the percentage of flow through the particle can be computed.

Figure 5.24(a) and (b) show the pressure drop across packed columns of POROS I and POROS II respectively. Three different sizes of the particles were used in these two cases - nominal diameters of 10 μm, 20 μm and 50 μm

particles. The actual mean diameters of the particles were obtained using laser scattering technique and the results are shown in Table 5.8 The standard deviation of the particle size distribution was calculated from SEM image analyses since the laser scattering particle size analyzer used is currently incapable of accurately the standard deviation of the particle size distribution.

Figure 5.25 shows a plot of the bed permeability, K_{p-s} as a function of the square of the particle diameter. Unfortunately, the data do not show the linearity as predicted by the Park and Stephanopoulos analysis. To determine the reason for the departure of the data from the expected linear relationship, the permeability of the packed columns was each calculated using the Blake-Kozeny equation (Equation (5.12)) and the values of these were compared to the experimental data. Table 5.9 shows a compilation of the results obtained.

$$K_{B-K} = \frac{u_b}{\Delta P/\mu L} = \frac{\varepsilon_b^3}{150(1-\varepsilon_b)^2} D_p^2 \qquad ...(5.12)$$

while the experimental values were calculated from $\frac{u_b}{\Delta P/\mu L}$.

From Table 5.9, for both POROS I and II, the calculated Blake-Kozeny permeabilities compare reasonably well with the experimental values for the 12 μm and 23 μm particles. As for the larger 54 μm particles, there is a large discrepancy between the two sets of data. This may be attributed in part to the difficulty of measuring the very low pressure differentials across the columns packed with the 54 μm particles. It was found that at the very low pressures, the Biocad can only measure pressure differentials to within 50% in error.

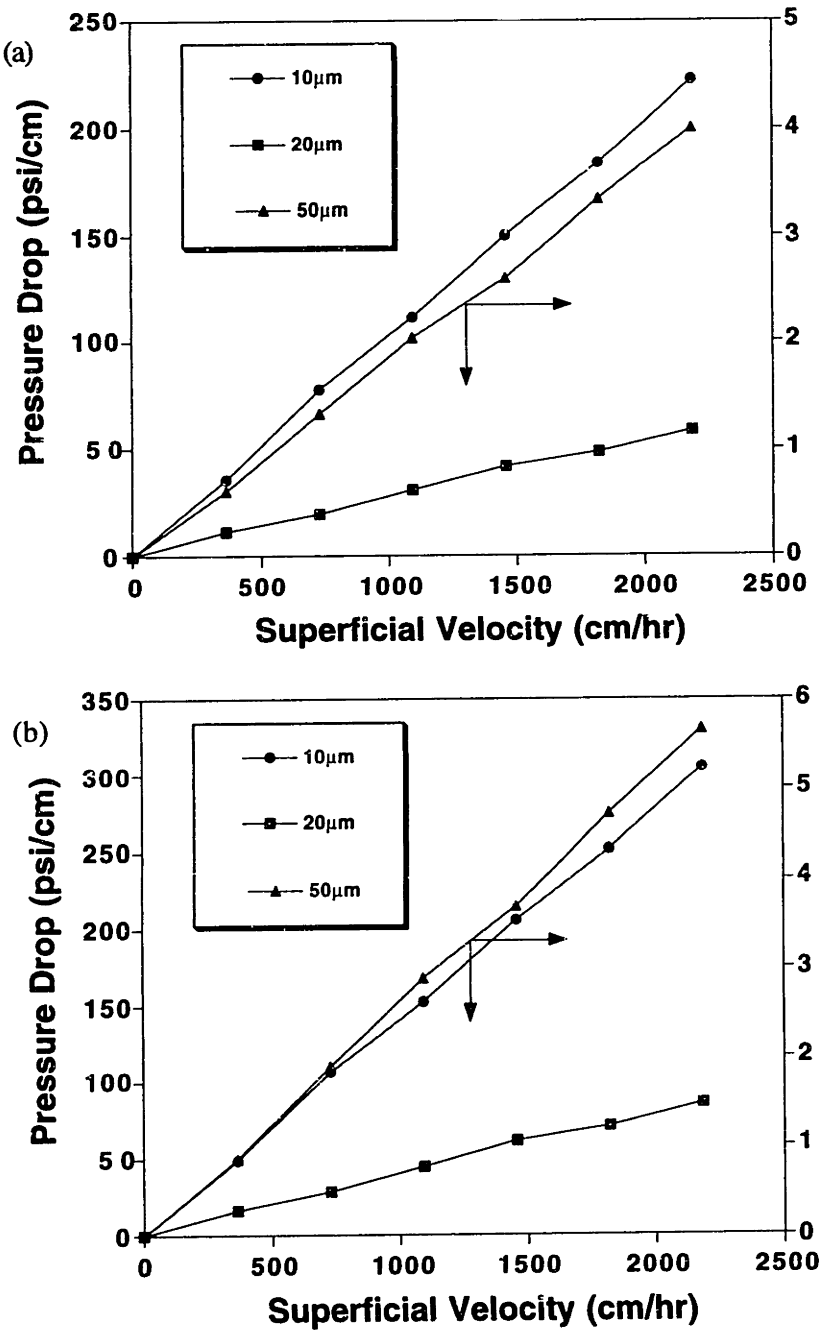


Figure 5.24 Pressure drop across packed columns of (a) POROS I and (b) POROS II for particles of different nominal diameters.

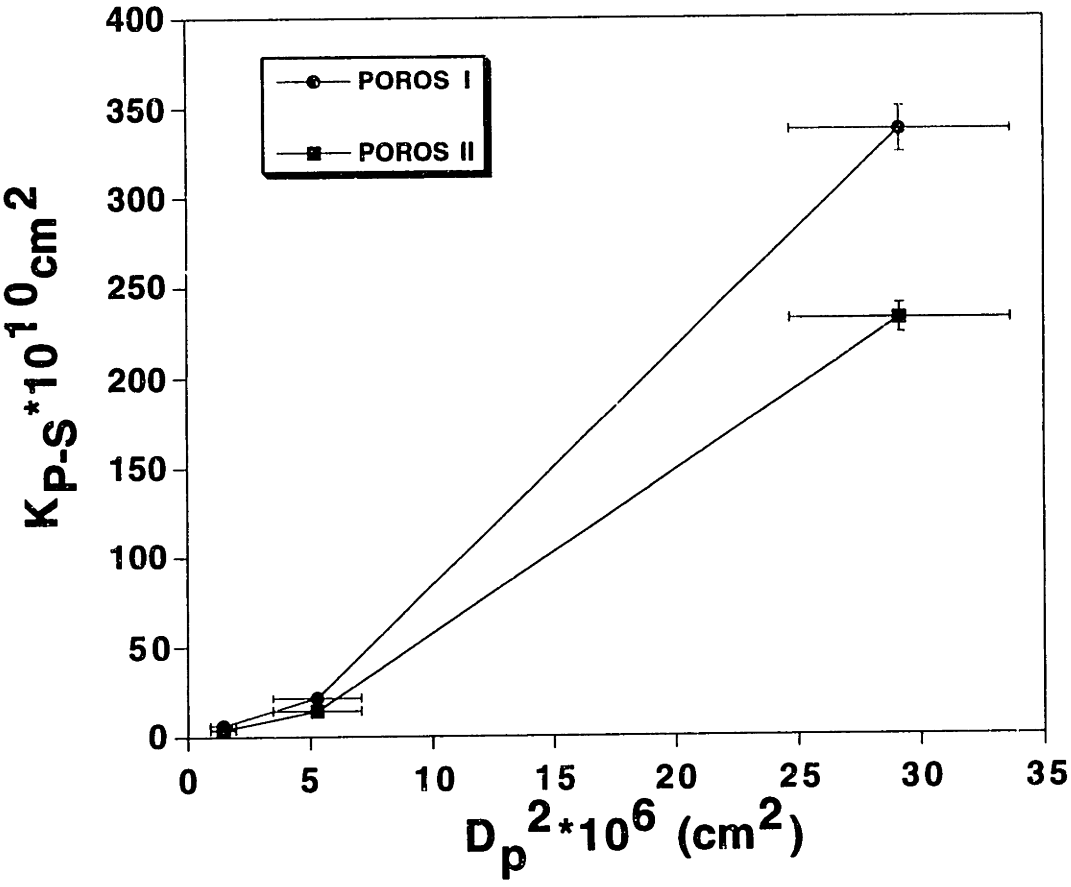


Figure 5.25 Plot of bed permeability as defined by Park and Stephanopoulos (1992) as a function of the square of the diameter of the packing particles.

Table 5.8 Particle Size Distributions of POROS Packing

The Blake-Kozeny equation is but one of many equations that relates the Both POROS I and II were found to have the same particle size distribution and hence, only parameters for POROS I are shown.

Nominal Particle Diameter (μm)	Mean Particle Diameter from Laser Scattering (μm)	Standard Deviation of Particle Diameter from SEM Image Analysis (μm)
10	12	2
20	23	4
50	54	5

Table 5.9 Comparison between Column Permeabilities Calculated using the Blake-Kozeny Equation with the Experimental Obtained Values

Particle Diameter (μm)	Permeability for POROS I ($x10^{10}$ cm^2)			Permeability for POROS II ($x10^{10}$ cm^2)		
	Blake-Kozeny	Expt.	Difference (%)	Blake-Kozeny	Expt.	Difference (%)
12	2.67	2.23	20	1.97	1.52	30
23	9.80	8.05	22	7.25	5.55	31
54	54.0	127	235	39.9	89.5	224

Consequently, the experimentally derived permeability is questionable. Another reason for the disparity may be the applicability of the Blake-Kozeny equation. The Blake-Kozeny equation is but one of many equations that relates the column permeability to the particle size and porosity of the packed bed. Much effort has been expended to find the best functions of particle size and porosity so as to be able to represent all the data on bed permeability by the same relationship. In fact, Carmen (1937) had proposed that the Kozeny constant, "a" (the number 150 in the denominator of the Blake-Kozeny equation), to be 180 (a difference of 20% from that used in the Blake-Kozeny equation). If this value were used in the Blake-Kozeny equation, the data (for the 12 μm and 23 μm particles) in Table 5.9 would be brought to an even closer reconciliation. The work of Rumpf and Gupte (1971), however, suggests that a general formula does not exist. The value of the Kozeny constant was found to depend on the particle size distribution parameters, and the porosity function in the permeability relationship may even be expected to depend on the particle shape and the packing structure (Probstein, 1989). The different forms that have been suggested for the porosity function, $f(\varepsilon_b)$, by different authors are shown in Table 5.10

Nevertheless, since the calculated permeabilities using the Blake-Kozeny equation only differ from the experimental values by less than 30% for the 12μm and 23 μm particles, the Park and Stephanopoulos analysis is deemed suitable for analysing intraparticle convection in these packings. Figure 5.26 is a replot of the bed permeability as a function of the square of the particle diameter for the 12 μm and 23 μm particles. Using linear regression, the relationship between K_{p-s} and D_p was found to be:

POROS I : $K_{p-s} = 0.127 + 4.034 D_p^2$ ...(5.13)

POROS II : $K_{p-s} = 0.0364 + 2.714 D_p^2$ ...(5.14)

Table 5.10 Different Porosity Functions for Low Re Number Flow (Adapted from Rumpf and Gupte (1971))

$1/f(\varepsilon_b)$	Author
$(1-\varepsilon_b)^2/\varepsilon_b^3$	Blake (1922), Kozeny (1927), Carmen (1937)
$(1-\varepsilon_b)^2/\varepsilon_b$	Zunker (1920)
$[(1-\varepsilon_b)^{1.3}/(\varepsilon_b - 0.13)]$	Terzaghi (1925)
$[(1.115(1-\varepsilon_b)/\varepsilon_b^{1.5}][(1-\varepsilon_b)^2 + 0.018]$	Rapier (1949)
$69.43 - \varepsilon_b$	Hulbert and Feben (1933)
$\varepsilon_b^{-3.3}$	Slichter (1898)
$\varepsilon_b^{-1.0}$	Kruger (1918)
$\varepsilon_b^{-6.0}$	Hatch (1934), Mavis and Wilsey (1936)
$\varepsilon_b^{-4.0}$	Fehling (1939)
$\varepsilon_b^{-4.1}$	Rose (1945)
$\varepsilon_b^{-5.5}$	Rumpf and Gupte (1971)

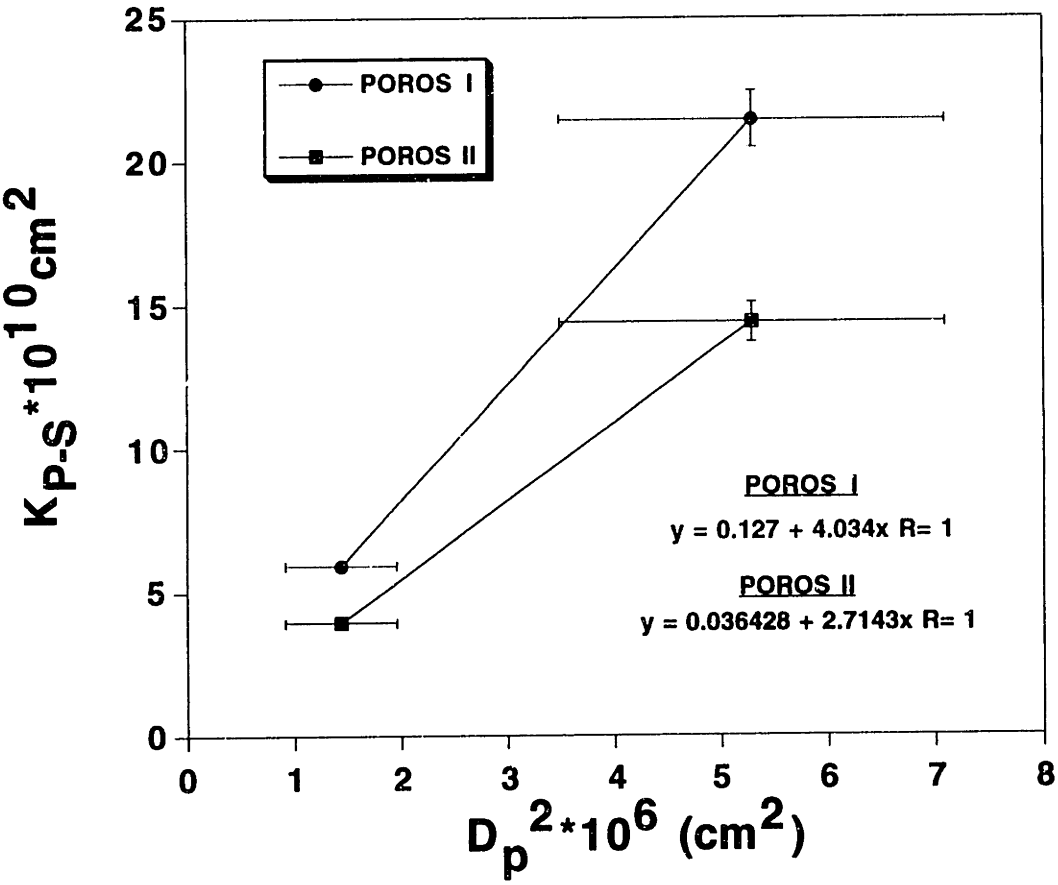


Figure 5.26 Replot of bed permeability as a function of the square of the diameter of the packing particles. The linear relationships shown in the graph were obtained from linear regression of the data.

Based on Equations (5.13) and (5.14), the fraction of liquid flow through the porous packing is represented by:

$$\text{POROS I} : \frac{Q_p}{Q_b} = \frac{1}{1+32D_p^2} \qquad ...(5.15)$$

$$\text{POROS II} : \frac{Q_p}{Q_b} = \frac{1}{1+75D_p^2} \qquad ...(5.16)$$

Using fractional error analysis, the fractional error in calculating the K_{p-s} from experimental data was computed to be less than 20%. Hence, for the case of the 23 μm packings, the amount of convective flow through the particles as a percentage of the total flow through the column can be calculated as:

$$\text{POROS I} : \frac{Q_p}{Q_b} = \frac{1}{1+32(2.3)^2}$$

$$= 0.59 \pm 0.12\%$$

$$\text{POROS II} : \frac{Q_p}{Q_b} = \frac{1}{1+75(2.3)^2}$$

$$= 0.25 \pm 0.05\%$$

The corresponding values of $0.60 \pm 0.06\%$ for POROS I and $0.23 \pm 0.02\%$ for POROS I as computed using the network model corroborate with the modified Blake-Kozeny analysis by Park and Stephanopoulos. To further confirm the validity of the transport methodology used with the network model, these results will be compared with the theoretical analysis proposed by Rodrigues and co-workers (1991).

5.10.3 Medium Permeability Based on Modified van Deemter Equation

Using a rigorous theoretical treatment based on the macroscopic continuity equations on the particle level and the packed column, Rodrigues and co-workers (1991) developed an extended van Deemter equation for the reduced plate height in an HPLC column. The extended van Deemter equation incorporates the effects of intraparticle convection in terms of a convection-augmented diffusivity concept. For a detailed discussion of their analysis, the reader is referred to the paper by Rodrigues *et. al.* (1991). The resulting set of equations from their analysis is summarized as follows:

$$\text{Reduced } HETP = \frac{\sigma^2}{\mu_1^2}\frac{L}{D_p} \qquad \text{...(5.17)}$$

$$\sigma^2 = \frac{2}{Pe}\mu_1^2 + \frac{2}{15}\varepsilon_p\frac{(1-\varepsilon_b)}{\varepsilon_b}\frac{1}{n_t}f(\frac{\lambda}{3}) \qquad \text{...(5.18)}$$

Here, σ^2 is the variance of the eluted peak,

μ_1 is the mean residence time of the eluted peak.

Pe is the Pe number in the packed bed given by Equation (5.19):

$$Pe = \frac{u_b L}{\varepsilon_b D_{ax}} \qquad \text{...(5.19)}$$

where u_b and ε_b have their usual definitions,

D_{ax} is the axial dispersion coefficient and

L is the length of the column

The number of transfer units for diffusion in the throughpores, n_t, is defined as

$$n_t = \frac{4\varepsilon_b L D_{eff}}{\varepsilon_p D_p^2 U_b} \qquad ...(5.20)$$

where D_{eff}, the pore diffusivity of the solute in the macropores is defined as

$$D_{eff} = \frac{\varepsilon_b D_e}{\tau} \qquad ...(5.21)$$

where D_e is the pore diffusivity of the solute, and

τ is the tortuosity factor

The pore diffusivity, D_e, is defined as the solute bulk diffusivity corrected for finite size of the solute. This is calculated from Brenner and Gaydos (1977) as:

$$D_e = D_m(1 - \frac{9}{8}\eta \ln \eta^{-1} - 1.539\eta) \qquad \text{for } \eta < \sim 0.1$$

where D_m is the solute bulk diffusivity, and

η is the ratio of the solute diameter to the macropore diameter

The function $f(\lambda)$ contains the contribution of intraparticle convection for the peak variance. It was first introduced by Rodrigues *et. al.* (1982) for measuring effective diffusivities in porous catalyst using a chromatographic technique. In the analysis, $f(\lambda)$ was obtained as :

$$f(\lambda) = \frac{3}{\lambda}(\frac{1}{\tanh \lambda} - \frac{1}{\lambda}) \qquad ...(5.22)$$

where $\lambda = \frac{u_p D_p}{2D_{eff}}$ and u_p the intraparticle velocity.

The axial dispersion, D_{ax}, in the column is calculated from the relationship suggested by Gunn (1971) based on stochastic theory:

$$\frac{D_{ax}}{u_b D_p} = [\frac{Pe_b}{\varepsilon_b \Gamma}(1-p)^2 + \frac{Pe^2}{\varepsilon_b^2 \Gamma^2}(1-p)^3\{\exp[\frac{-\varepsilon_b \Gamma}{p(1-p)Pe_b}] - 1\}](\sigma_b^2 + 1) + \frac{\sigma_b^2}{2} + \frac{\varepsilon_b}{1.4 Pe_b}$$

...(5.23)

where Pe_b is the bed Pe number, given by $Pe_b = \frac{u_b D_p}{D_m}$

$\Gamma = \frac{4(1-\varepsilon_b)\alpha_1^2}{\varepsilon_b}$ where α_1 = first zero of the Bessel function , $J_0(x) = 0$

p = probability of axial displacement, defined as $p = 0.17$ for Re<1,

σ_b^2 = variance of velocity distribution over the cross-section

The velocity distribution variance, σ_b^2, is a measure of the quality of the packing in the column. For a well-packed bed, σ_b^2 will be small but a badly packed bed will result in large σ_b^2.

For the RTD analysis of lysozyme on the perfusive columns, the values of the parameters necessary to compute the reduced HETP are summarized in Table 5.11. Values of τ , σ_b^2 and u_p can only be determined using a multivariable regression technique given a set of experimental data for reduced HETP as a function of superficial velocity, u_b. Using the experimental plate height data obtained for the columns packed with POROS I, II and the blends of POROS I and II, τ, σ_b^2 and u_p were evaluated using the SOLVER non-linear multivariable regression subroutine in the computer program, MICROSOFT EXCEL. In the case of u_p, this was obtained as $K_u = u_p/u_b$ = ratio of intraparticle convective velocity to superficial bed velocity.

Figures 5.27 and 5.28 show the comparison between the calculated and the experimental HETP after the non-linear regression. The solid line represents the equation y = x. The fit between calculated and experimental

values is excellent in all cases. The results of the regression analysis are summarized in Table 5.12.

Table 5.11 Parameters used in Calculating the Reduced HETP Based on the Extended van Deemter Equation (Equation 5.13)

Parameter	Value
L(cm)	10
ε_b	POROS I: 0.25 POROS II: 0.23
ε_p	POROS I: 0.57 POROS II: 0.61
$D_m(cm^2/s)$	2.19×10^{-6}
$D_p(cm)$	23×10^{-4}
α_1	2.4048

It is appropriate, at this point, to address the effect of the low column porosity on the split ratio obtained from the multivariable regression analysis. As mentioned in Section 5.2, there is a discrepancy in the column porosity obtained from back calculation. The column porosity, ε_b, was changed to 0.30 in both cases of POROS I and II, with all the other parameters kept constant, and the split ratios of velocity and flow rate were evaluated using the multivariable regression technique. It was found that for POROS I, the change in the split ratio of the velocity was 3% (from 0.0078 to 0.0080) and the change

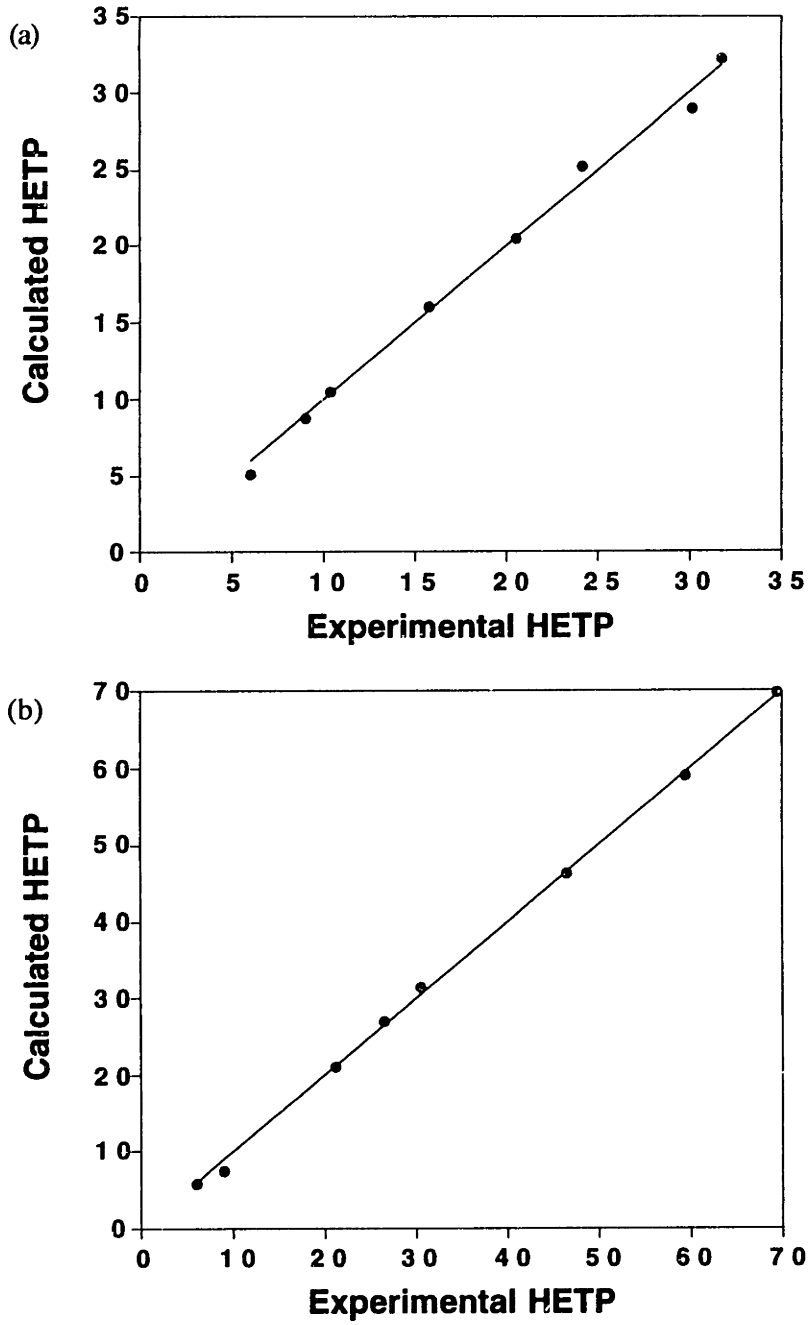
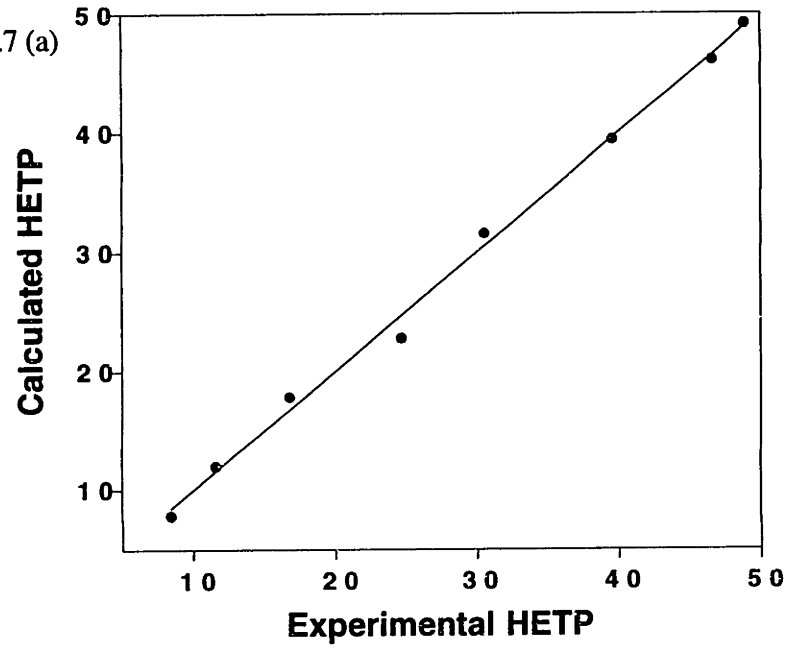
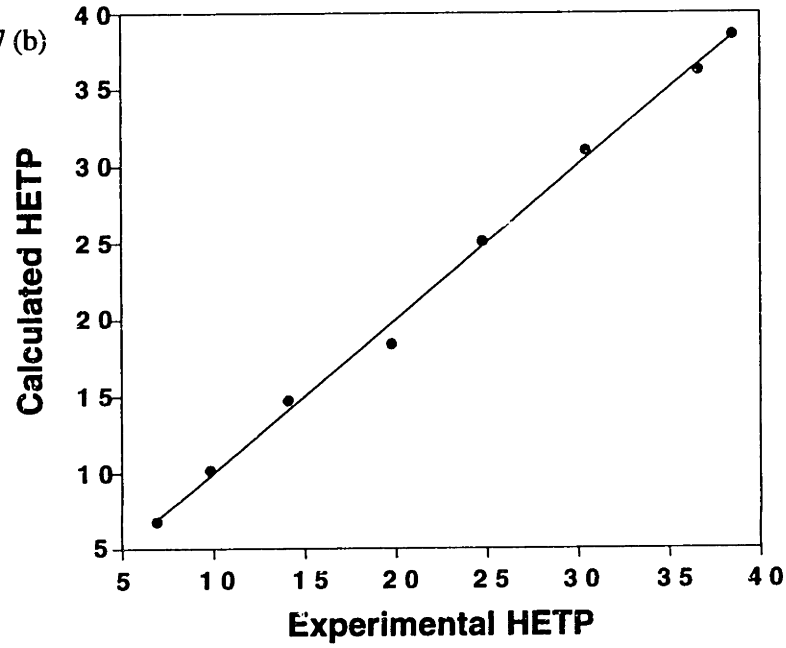


Figure 5.27 Comparison between experimental and calculated HETP. Calculated HETP was obtained from nonlinear fit to experimental data. (a) POROS I (b) POROS II

5.7 (a)


Calculated HETP
50
40
30
20
10
10
20
30
40
50
Experimental HETP

5.7 (b)


Calculated HETP
40
35
30
25
20
15
10
5
5
10
15
20
25
30
35
40
Experimental HETP

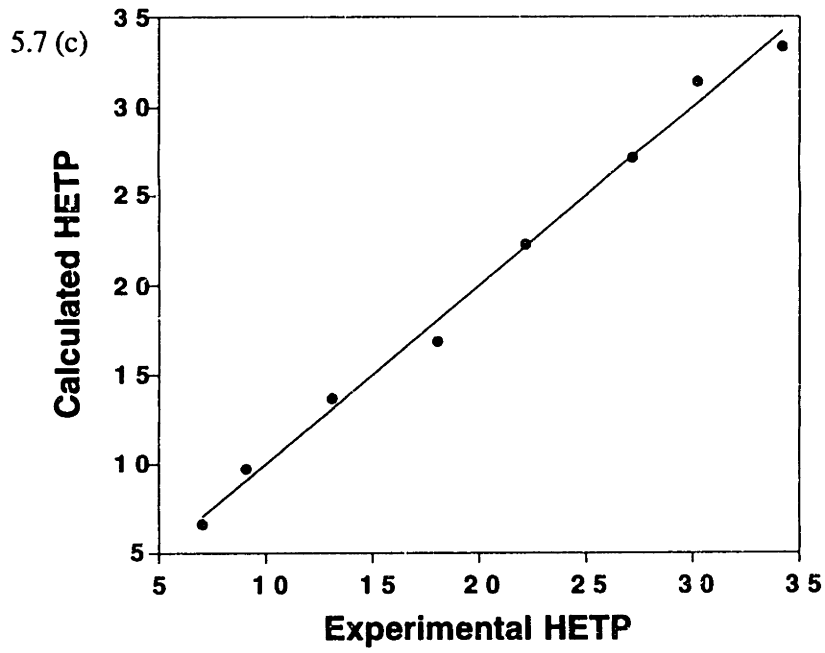


Figure 5.28 Comparison between experimental and calculated HETP. Calculated HETP was obtained from nonlinear fit to experimental data. (a) 25/75 POROS I/II (b) 50/50 POROS I/II (c) 75/25 POROS I/II

in the split ratio of flow rate was 2% (from 0.57% to 0.56%) in response to a change in the column porosity by 20% (from 0.25 to 0.30). In the case of POROS II, the change in the split ratio of the velocity was 13% (from 0.0030 to 0.0026) and the change in the split ratio of flow rate was 18% (from 0.23% to 0.19%) in response to a change in the column porosity by 30% (from 0.23 to 0.30). Therefore, it can be concluded that the split ratios of intraparticle convection are not very sensitive to changes in the column porosity, and the discrepancy in the back calculations for the column porosity is consequently not significant.

Table 5.12 Results of Regression Analysis from Experimental Plate Height Data

Parameter	POROS I	25/75 II/I	50/50 I/II	75/25 II/I	POROS II
Tortuosity factor, τ	1.22	1.46	1.74	2.35	2.91
Packing Quality, σ_b^2	3.02	1.04	1.00	1.18	3.09
$K_u = u_p/u_b$ $x10^3$	7.81	5.73	4.85	3.97	3.04
Q_p/Q_b (%)	0.57	0.42	0.36	0.30	0.23

As the macropore sizes in the packings increase from 3500Å (POROS II) to 5000Å (POROS I), the tortuosity factor obtained decreases from 2.91 to 1.22. This is often observed for porous media and probably reflects a decreasing number of dead-end pores as the pores become larger. This is consistent with

data reported by Frey *et. al.* (1993). By matching experimental and theoretical results of plate height data for various gigaporous packings, they have found that the tortuosity factor decreased from 5.1 to 3.1 for pore sizes ranging from 1000Å to 5000Å respectively. The packing quality of the columns were reasonable within the range stated by Gunn (1971). He had stated that when particles in the packed bed are very fine (in the micron range) and the flow in the bed is of very low Re number, σ_b^2 may rise as high as 1-3. In comparison, for experimental data of axial dispersion reported by Miller and King (1966), at low Re numbers and particles ranging in size down to 50 μm, values of σ_b^2 as large as 2.0 were suggested. The σ_b^2 values obtained for the perfusive columns are also consistent since the three blended columns were packed on the same day and it is expected that their packing quality will be similar. As for the columns packed with POROS I and POROS II, their packing qualities are also comparable.

Table 5.13 summarizes the percentage of intraparticle convective flow as computed from the network model and the two theoretical macroscopic approaches by Park and Stephanopoulos and Rodrigues *et. al.* The agreement in the results between the values computed using the network model and the theoretical approaches is remarkable. This corroboration provides an excellent validation for the transport methodology for elucidating the magnitude of intraparticle convective flow in perfusive media using the developed network model. Although the extended van Deemter equation reported by Rodrigues *et. al.* (1992) is appropriate for analyzing column efficiency in the presence of intraparticle convection, the method suffers from the disadvantage of requiring the values of empirical factors of tortuosity and the packing quality of the packed bed. These parameters, which are characteristics of the PSD of the packing

material and the packed column cannot be determined *a-priori*. Their method, nonetheless, is an excellent approach for elucidating the magnitude of the intraparticle convective flow given experimental data of HETP as a function of superficial bed velocity. In fact, combining the microscopic network model with the macroscopic theoretical model will provide an excellent characterization of perfusive medium.

With the validation of the transport methodology for the network model, it becomes possible to study the influence of the various parameters describing the PSD of the porous medium on intraparticle convection. This is the subject of discussion in the next section.

Table 5.13 Summary of Intraparticle Convective Flow Rate Determined from Different Approaches

Intraparticle Convective Flow as a % of Total Flow Through Packed Bed			
Column	Network Model	Rodrigues *et. al.* (1991) Extended van Deemter Equation	Park and Stephanopoulos (1992) Modified Blake-Kozeny Equation
POROS II	0.23	0.23	0.25
25/75 I/II	0.31	0.30	---
50/50 I/II	0.38	0.36	---
75/25 I/II	0.41	0.42	---
POROS I	0.60	0.57	0.59

5.11 Effects of PSD and PI on Intraparticle Convection

While the macroscopic models developed by Park and Stephanopoulos (1992) and Rodrigues *et. al.* (1991) can be used to study the effects of intraparticle convection on column efficiency, these models cannot account for the importance of pore morphology and topology on the magnitude of intraparticle convection. In the case of Park and Stephanopoulos, it has been assumed that the Blake-Kozeny equation is suitable for modelling permeability through both the packed bed and the packings. Nevertheless, it is still unclear as to whether the Blake-Kozeny equation is appropriate for an interconnected pore network in a porous medium (Dullien, 1992). Many researchers (Dullien, 1992; Rumpf and Gupte, 1971; Probstein, 1989) nave ascertained that the relationship between permeability and particle size is dependent on the particle size distribution (or pore size distribution) and a complicated function of porosity (refer to Section 5.10.2).

In another macroscopic model, Rodrigues and his co-workers (1991) have incorporated the effects of the pore morphology and topology into the tortuosity factor in their model. It has been known now that the tortuosity factor is a strong function of the pore size distribution and the pore interconnectivity of the porous medium (Portsmouth and Gladden, 1992; Reyes and Jensen, 1987; Sahimi, 1991; and others). The results from macroscopic modelling are not sufficient to predict the effects of the PSD and PI on intraparticle convection.

The network model developed in this research sought to evaluate intraparticle convection on a microscopic level by taking into account the effects of changing parameters of the PSD and PI. Here, the parameters characterizing

the PSD and the PI were varied and the effects of such variations on the magnitude of intraparticle convection were studied. In essence, a "computer experiment" was performed for each change in the parameter of the PSD. Each computer experiment involved generating the network using the new PSD and/or PI, and computing the split ratio of the flow rate through the particle using the transport methodology that has been validated in the previous sections. the split ratio of the flow rate is defined as the intraparticle flow rate as a percentage of the total flow rate through the bed (Q_p/Q_b).

The parameters which are of interest here are the mean sizes of the macropores and micropores, which were rendered dimensionless by taking the ratio of these to the particle diameter and the standard deviation of the sizes of the macropores and micropores, which were non-dimensionalized by division with the mean macropore diameter and mean micropore diameter respectively. In addition, the effect of PI was also studied. By understanding the relationships between the PSD and PI and the magnitude of intraparticle convective flow, the porous structure of the packing can be systematically designed and optimized.

The parameters were varied in such a manner that the porosities of the bed and the particle were kept constant. Since the network lattice is finite and limited in size, it was not possible to simultaneously maintain the surface area in the network as this would require numbers of pores which are magnitudes larger than can be simulated in the lattice. Hence, the proportion of macropore to micropore was varied to maintain the porosities. For each computer experiment, a total of 75 realizations were carried out to reduce the standard deviation of the split ratio to less than 5% of the mean values in the simulations.

5.11.1 Effects of Pore Interconnectivity (PI)

The PI was varied from 2.44 to 4.8 in generating the networks to represent the porous structures of POROS I, POROS II and blends of POROS I and II. This corresponded to varying the fraction of occupied pores in the lattice from 0.40 to 0.80. Figure 5.29 shows the split ratio of flow rate through the particle as a function of PI for the various media. For each medium, it was found that a twelve-fold increase in the amount of flow rate through the particles can be achieved when the PI was increased two-fold. As the interconnectivity between the pores in the particles increases, more open paths are available for liquid flow. Moreover, on the level of the packed bed, since the number of pores exposed to the surface of the particles is directly proportional to the PI, a greater flow rate through the particles is also expected with higher PI. It is not surprising, therefore, that the split ratio should increase with PI. In addition, the data indicate that the relationship between the split ratio and PI seems to follow a power-law and linear regression of the data was carried out to evaluate this relationship. Table 5.14 tabulates the regressed slopes and the correlation coefficients. An average slope of 3.67 ± 0.02 was obtained. Hence, $Q_p/Q_b \sim (PI)^{3.67}$.

Table 5.14 Regression Parameters for Effects of PI on Split Ratio

Medium	Slope	Correlation Coefficient, r^2
POROS I	3.67	0.97
25/75 I/II	3.64	0.99
50/50 I/II	3.69	0.99
75/25 I/II	3.68	0.98
POROS II	3.68	0.97

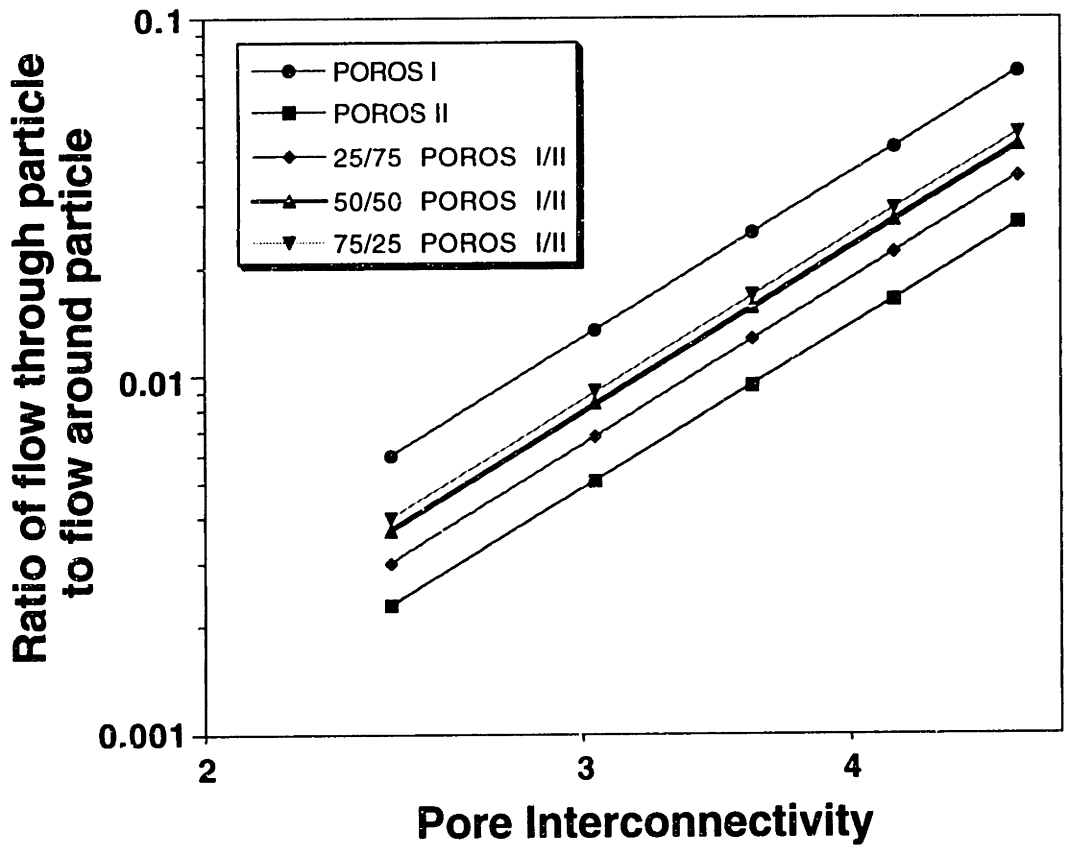


Figure 5.29 Effects of pore interconnectivity on the split ratio of flow rates. Each datum is the average of 75 realizations.

5.11.2 Effects of Macropore Diameter

The impact of the diameter of the macropores in the PSD was studied by varying the dimensionless mean macropore diameter, $d_{ma} = d_{macro}/D_p$, where d_{macro} is the diameter of the macropore and D_p is the diameter of the particle. The base cases examined were POROS I, POROS II and the 25/75 POROS I/II blend. For each base case, d_{ma} was varied while the remaining parameters of the PSD, namely the standard deviation of the macropore distribution, the parameters of the micropore size distribution, the particle size distribution and the PI, were kept constant. Table 5.15 tabulates the ranges of d_{ma} and the corresponding mean macropore diameter studied in each base case.

Table 5.15 Range of Reduced Macropore Diameter used for Studying the Effects of Macropore Diameter

Base Case	Range of Reduced Macropore Diameter	Corresponding Mean Macropore Diameter
POROS I	0.0217-0.0257	5000-5900 Å
POROS II	0.0152-0.0185	3500-4250 Å
25/75 I/II	0.0183-0.0222	4200-5100 Å

The results of the split ratio of flow rate plotted as a function of the dimensionless macropore diameter are shown in Figure 5.30. As the mean macropore size is increased, the percentage of the total flow rate through the bed that convects through the particle also increases. Straight line relationships are discernible from the results in Figure 5.30. Regression of the data provides $Q_p/Q_b \sim (d_{macro}/D_p)^{4.74}$ with an average correlation coefficient of $r^2=0.98$.

5.11.3 Effects of Micropore Diameter

Similar to the mean macropore diameter, the mean micropore diameter was non-dimensionalized by division with the particle diameter, that is, $d_{mi} = d_{micro}/D_p$, where d_{micro} is the mean diameter of the micropores. The base cases investigated were again POROS I, POROS II and the 25/75 I/II blend. Table 5.16 shows the ranges of mean micropore diameter studied for each case.

Table 5.16 Range of Dimensionless Mean Micropore Diameter used for Studying the Effects of Macropore Diameter

Base Case	Range of Reduced Micropore Diameter x 10^3	Corresponding Mean Micropore Diameter
POROS I	1.087-1.348	250-310Å
POROS II	0.696-0.957	160-220Å
25/75 I/II	0.739-1.043	170-240Å

Figure 5.31 shows the results of the computer experiments for varying dimensionless micropore diameters. It can be seen that the split ratio through the particle is only marginally dependent on the size of the micropore. Regression of the data reveals that $Q_p/Q_b \sim (d_{micro}/D_p)^{0.21}$. This is consistent with the assumptions made by the various researchers of macroscopic modelling (Frey *et. al.*, 1993; Rodrigues *et. al.*, 1991; Carta and Rodrigues, 1994) that intraparticle convection only occurs in the macropores while diffusion takes

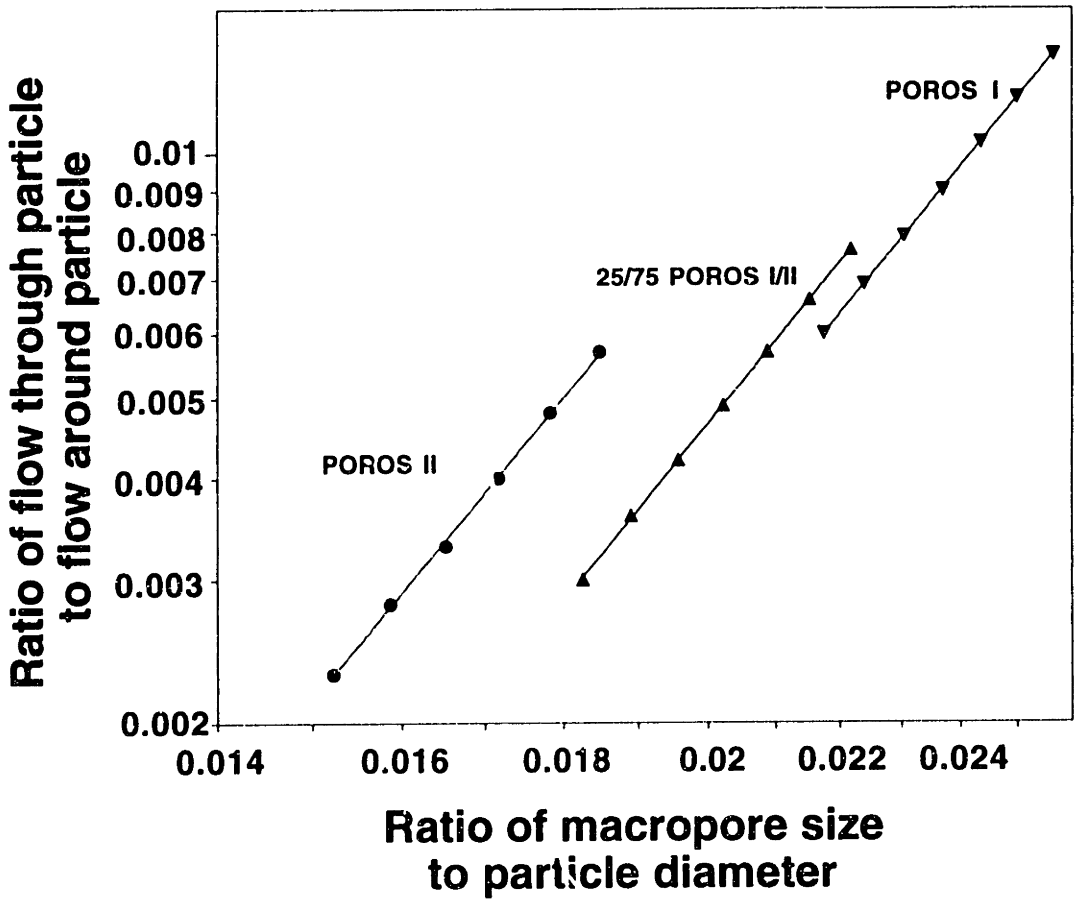


Figure 5.30 Effects of diameter of macropore on the split ratio through the particle. The macropore diameter is non-dimensionalized by division with the particle diameter. Base cases shown are POROS I, POROS II and 25/75 POROS I/II. 75 realizations were undertaken to obtain the average split ratio at each datum.

place in the micropores. Hence, changing the mean size of the micropore has very little effect on the amount of intraparticle convection.

5.11.4 Effects of the Spread of the Macropore Size Distribution

The effects of the spread of the macropore size distribution was investigated by changing the standard deviation of the macropore size distribution. The ratio of the standard deviation of the macropore distribution to the mean diameter of the macropore was varied for different base cases. Table 5.17 shows the ranges of the parameter examined for each base case.

Table 5.17 Range of Dimensionless Standard Deviation of Macropore Distribution used for Studying the Effects of the Spread of the Macropore Size Distribution

Base Case	Range of Reduced Standard Deviation of Macropore	Corresponding Range of Standard Deviation
POROS I	0.34-0.42	1700-2100Å
POROS II	0.40-0.50	1700-2100Å
25/75 I/II	0.49-0.60	1700-2100Å

Results of the computer simulations are shown in Figure 5.32. The results indicate that increasing the spread of the macropore size distribution has a monotonically increasing effect on the split ratio. As the standard deviation of the macropore distribution is increased for a fixed mean macropore diameter, the corresponding range of sizes of the macropore diameters also increase.

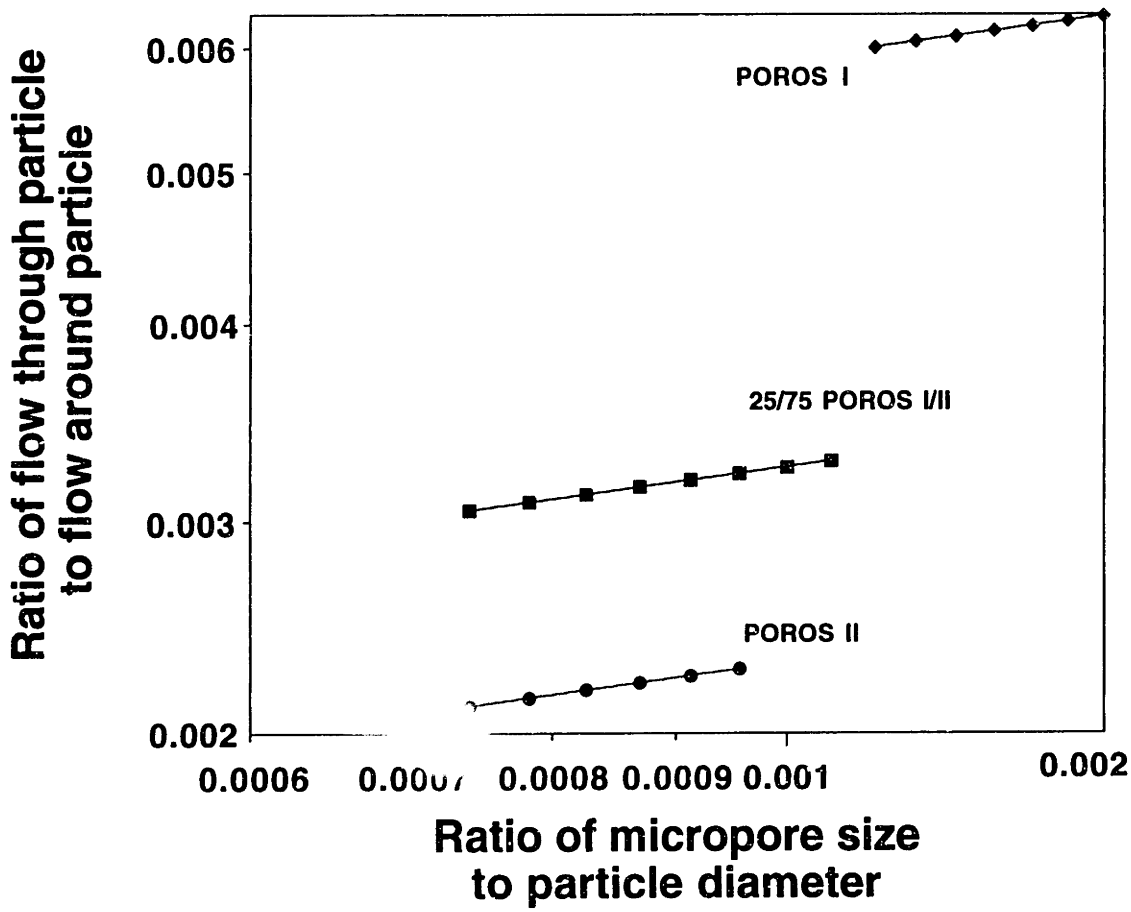


Figure 5.31 Effects of diameter of micropore on the split ratio through the particle. The micropore diameter is non-dimensionalized by division with the particle diameter. Base cases shown are POROS I, POROS II and 25/75 POROS I/II. 75 realizations were carried out to obtain the average split ratio at each datum.

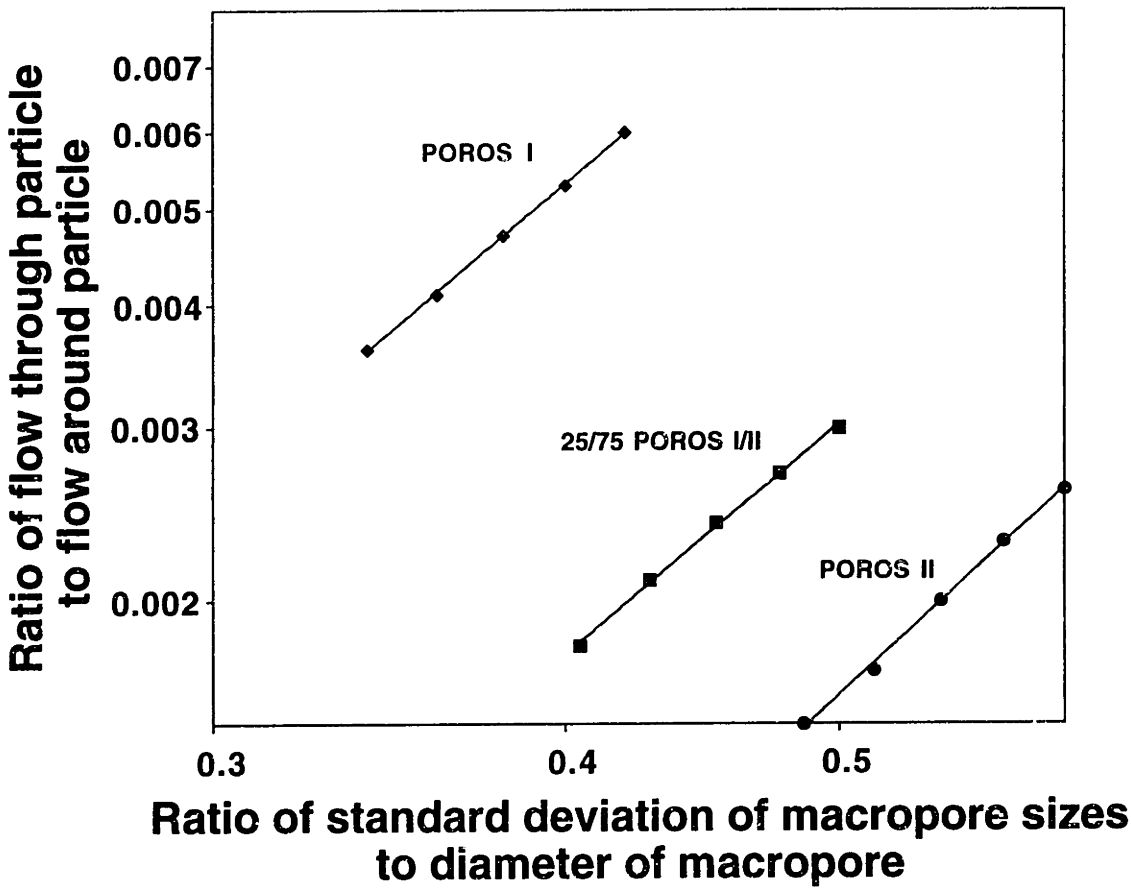


Figure 5.32 Effects of spread of macropore size distribution on the split ratio through the particle. The standard deviation of the macropore size distribution is non-dimensionalized by division with the macropore diameter. Base cases shown are POROS I, POROS II and 25/75 POROS I/II. 75 realizations were performed for each datum.

Consequently, the amount of intraparticle convection should elevate since it has been previously shown that intraparticle convection varies exponentially with the size of the macropores. Regression of the data gives an average slope of 2.50±0.13 with an average correlation coefficient of 0.98. Therefore, $Q_p/Q_b \sim (\sigma/d)^{2.50}_{macro}$.

5.11.5 Effects of the Spread of the Micropore Size Distribution

For studying the effects of the spread of the micropore size distribution, the standard deviation of the micropore size distribution was non-dimensionalized by the mean micropore diameter. The dimensionless standard deviation of the micropore distribution was varied and the split ratio Q_p/Q_b was computed for each of the resultant lattice realization. Table 5.18 lists the ranges of the dimensionless standard deviation investigated in the simulations.

Figure 5.33 plots the effects of the spread of the micropore size distribution on the magnitude of intraparticle convective flow. These results show that increasing the spread of the micropore distribution has an inverse effect on the split ratio through the particle. When the standard deviation of the micropore size distribution is increased, the range of micropore diameters in the composite PSD increases concomitantly. In order to maintain the same particle porosity, the proportion of macropores to micropores has to be decreased as a result. With a reduction in the relative numbers of macropores, the convective throughflow in the medium is subsequently reduced. In addition, decreasing the relative numbers of macropores increases the pore shielding effects of the macropores by the micropores. This pore-shielding effect results in a further reduction of the intraparticle convective flow. The results shown in Figure 5.33

indicate that the split ratio of the flow rate through the particle varies as $(\sigma/d)_{micro}^{-1.50\pm0.02}$ with an average correlation coefficient, r^2=0.98.

Table 5.18 Range of Dimensionless Standard Deviation of Micropore Distribution used for Studying the Effects of the Spread of the Micropore Size Distribution

Base Case	Range of Reduced Standard Deviation of Micropore	Corresponding Range of Standard Deviation
POROS I	1.68-1.92	420-480Å
POROS II	1.38-2.13	220-340Å
25/75 I/II	1.88-2.24	320-380Å

5.11.6 Overall Correlation for the Split Ratio

Based on the computer simulations which had been performed for varying parameters of the PSD and PI, it was found that the split ratio of the intraparticle convective flow rate can be approximated, at constant porosities of the bed and the particle by the empirical correlation:

$$\frac{Q_p}{Q_b}(\%) \sim (PI)^{3.67}\left(\frac{d_{macro}}{D_p}\right)^{4.74}\left(\frac{d_{micro}}{D_p}\right)^{0.21}\left(\frac{\sigma}{d}\right)_{macro}^{2.50}\left(\frac{\sigma}{d}\right)_{micro}^{-1.51} \qquad ...(5.24)$$

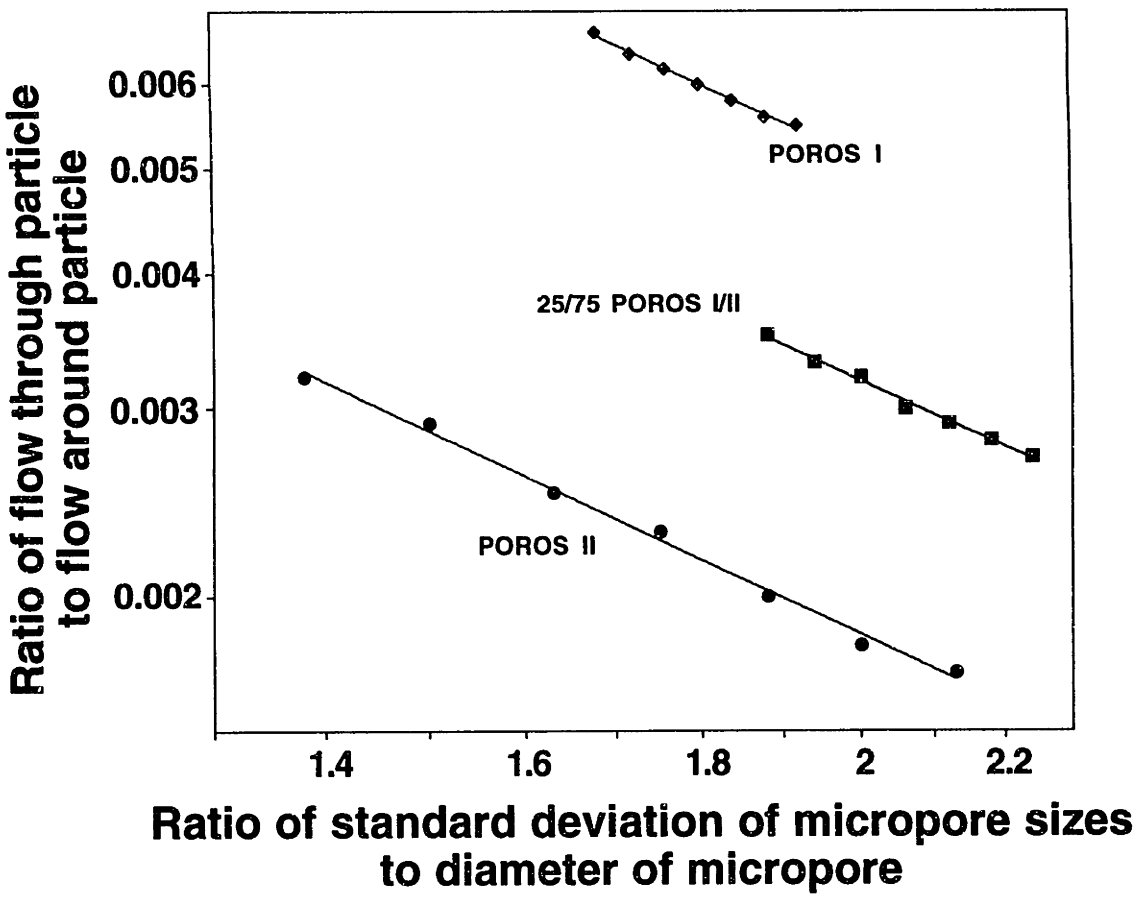


Figure 5.33 Effects of the spread of micropore size distribution on the split ratio through the particle. The standard deviation of the micropore size distribution was made dimensionless by division with the micropore diameter. Base cases shown are POROS I, POROS II and 25/75 POROS I/II. A total of 75 realizations were computed to obtain each average datum.

The strongest dependence of the split ratio can be seen to be related to the macropore size distribution: the mean diameter of the macropores and the standard deviation of the macropore size distribution. The pore-interconnectivity also plays a significant role in determining intraparticle convection.

The micropore size distribution, on the other hand, affects the magnitude of the intraparticle convection only marginally in terms of the mean micropore size and only moderately in terms of the variance of the micropore distribution. These results, therefore, reinforce the notion that in a perfusive packing having a bimodal pore size distribution, intraparticle convection occurs mainly in the larger macropores while the diffusive micropores provide additional adsorption capacity but do not contribute to intraparticle convective flows. Nonetheless, the micropore size distribution should be tightly controlled, and one should minimize the variance of the diameters of the micropores.

To regress all of the simulation data into one plot, the parameter, Q_{red}, given by Equation (5.25) was plotted against d_{ma} for all the data obtained from the computer simulations.

$$Q_{red} = \frac{(Q_p/Q_b)}{(PI)^{3.67}\left(\frac{d_{micro}}{D_p}\right)^{0.21}\left(\frac{\sigma}{d}\right)^{2.50}_{macro}\left(\frac{\sigma}{d}\right)^{-1.51}_{micro}} \qquad ...(5.25)$$

Figure 5.34 shows the regression fit for all of the simulation data against the empirical correlation given by Equation (5.24). It can be seen that the empirical correlation developed fits all the data to an average error of only about 0.7%. For comparison, experimental data for POROS I, POROS II and the three blends used in this work are also shown. The parameters of the PSD used

for these data were those obtained from the mercury porosimetry simulations. It is evident that the correlation provides excellent agreement with the experimental data.

Unfortunately, it is not possible to compare the results of the empirical correlation with published literature. Despite the fact that the characterization of the porous structure is of utmost importance to elucidate the effects of pore morphology and topology on the intraparticle transport phenomena, to date, no work has been done in this respect. All of the published work has lumped the size distribution of the pores into a tortuosity factor and a porosity function. This in turn results in the series of equations which relate the permeability of a packed bed with the particle diameter, and the Blake-Kozeny equation being one of those equations. Table 5.10 lists other related equations in this regard.

The power-law exponent of the split ratio of flow rate with the pore interconnectivity (PI) of 3.67 is consistent with the the scaling laws in percolation theory. Based on scaling theory, the conductivity factor, which is analogous to medium permeability in this research, obeys a power law relationship with the percolating fraction (which is directly proportional to PI) in the simulation lattice. Mohanty *et. al.* (1982) have reported that the power law exponent for a three dimensional cubic lattice near the percolation threshold should be greater than one. At fractions away from the percolation threshold, the simulated results reported revealed an exponent of approximately four, which compares very well with the exponent of 3.67 found here.

The exponent of the dimensionless macropore diameter of 4.74, however, is significantly different from that predicted by the Blake-Kozeny

exponent of two. This is hypothesized to be due to the network model used in the modelling. The microscopic parameters of the pore size distribution (the means and standard deviations of the macropore and micropore pore size distributions) were also varied as independent parameters. Due to the incorporation of these parameters, the exponent of the dimensionless macropore diameter can be much greater than two due to the effect of the macropore variances. In order to verify this postulation, the split ratio of intraparticle convection was plotted as a function of the reduced macropore diameter for the experimental data obtained for POROS I, POROS II and the blends investigated in this research and are shown in Figure 5.35. In this figure, the pore size distributions characterizing the porous structures of these media were not included. A linear regression of the data in Figure 5.35 indicates that $Q_p/Q_b \sim (\frac{d_{macro}}{D_p})^{2.58}$ with a correlation coefficient of r^2=0.94. The exponent of 2.58 was found to differ only slightly from the Blake-Kozeny exponent of two. This difference of about 30% hence corroborates the earlier explanation as to the larger exponent of 4.74 when the parameters of the pore size distribution were included in the analysis.

Figure 5.36 shows a replot of the experimental bed permeability data obtained from pressure drop calculations. It has been mentioned in Section 5.10.2 that part of the discrepancy in the permeability of the 54μm packing may be due to the interpretation differences of the Blake-Kozeny equation. Even when the Blake-Kozeny equation is applicable, there is considerable debate as to which particle diameter to use (see Dullien, 1992). Another note of caution when using the Blake-Kozeny equation is the potential errors at high bed porosity. At high porosities, it has been found (Dullien, 1975) that the concept of conduit flow (on which the Blake-Kozeny is based) breaks down and

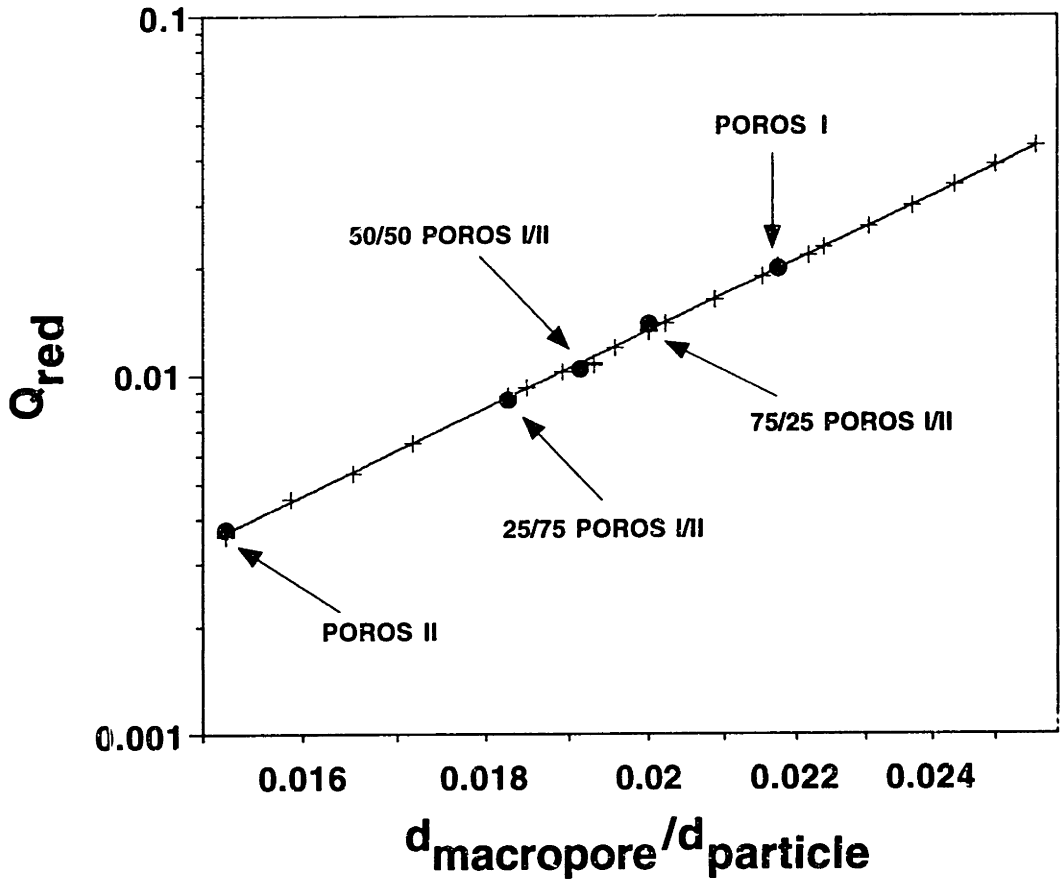


Figure 5.34 Regression fit of all the simulation data (+). Experimental data for POROS I, II and the blends used in this research are also shown for comparison (•). The solid line represents the reduced split ratio correlation of Equation (5.25).

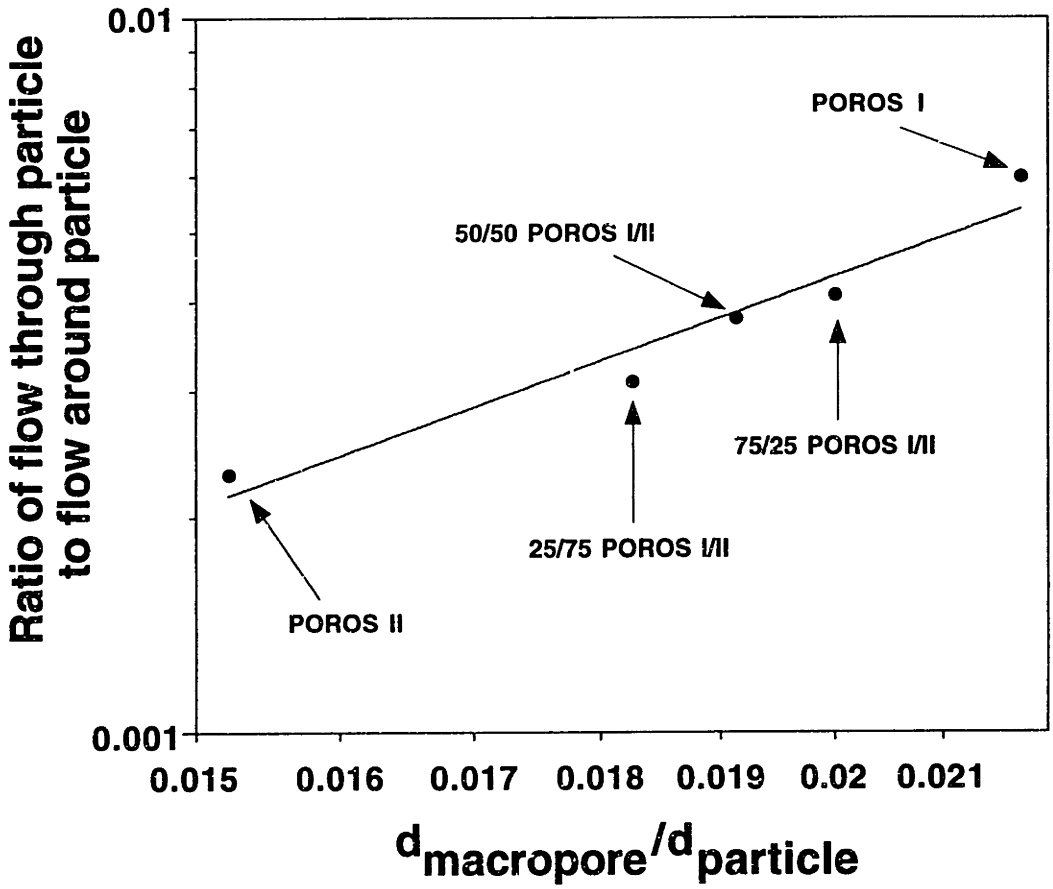


Figure 5.35 Plot of split ratio of intraparticle convection as a function of reduced macropore diameter. Data (•) represents experimental data obtained for POROS I, II and the various blends while the solid line represents a linear regression of the data: slope = 2.58 and r^2 = 0.94.

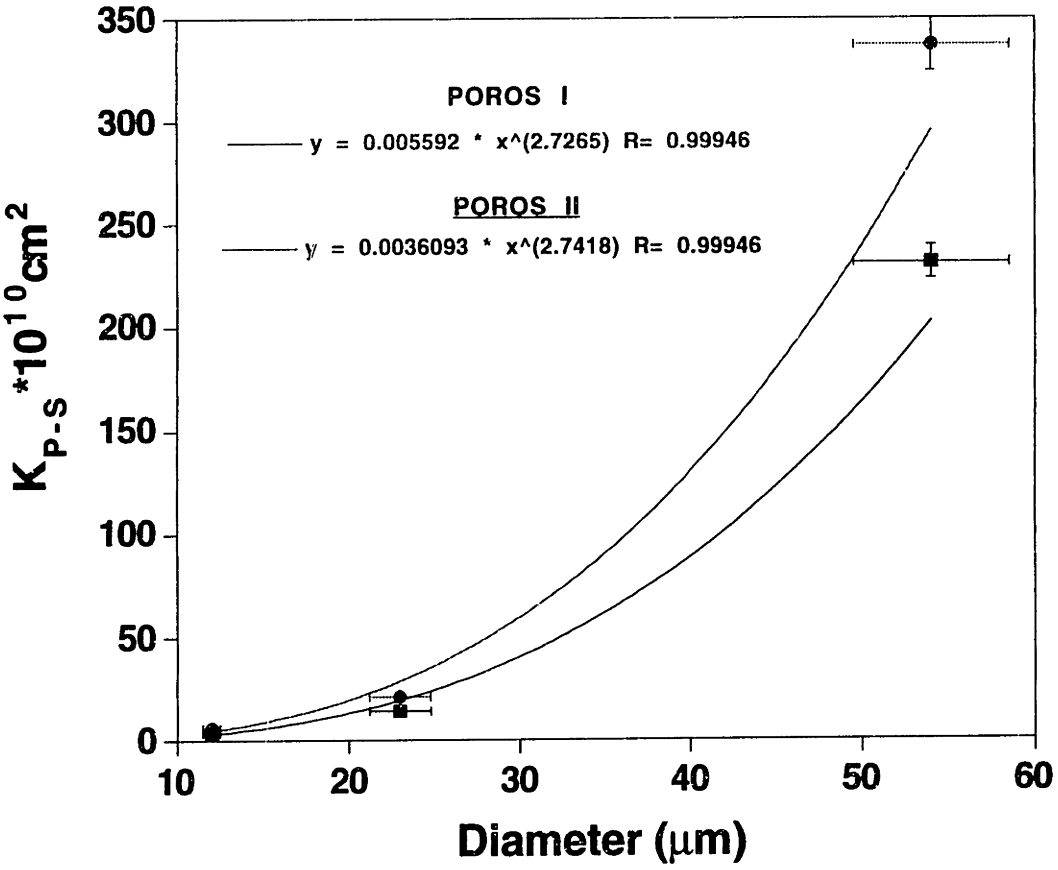


Figure 5.36 Replot of the experimental data of Figure 5.25. Non-linear regression was performed on the data and the regressed equations and correlation coefficients are as shown in the graph.

experimental data are better described by the approach of "flow around submerged objects". In the case of an interconnected pore network, pore non-uniformities (due to distribution and interconnectivity of pores) are also reasons for the disagreement between permeabilities calculated using the Blake-Kozeny equation and experimental values. In fact, the disagreement can differ by as much as 200% even when only pores of two different diameters (size ratio of 3:1) are present in the network (Dullien, 1975). By performing a non-linear regression on the experimental data in Figure 5.36, it was found that $K_{p\text{-}s} \sim (D_p)^{2.73}$. The exponent of 2.73 differs from that obtained in Figure 5.35 by about 6% but differs from the Blake-Kozeny exponent of 2.0 by 37%. From these analyses, it can be concluded that it is insufficient to correlate permeability data with just the mean pore size of the porous structure if systematic design and optimization of the porous structure is desired. On the other hand, the empirical correlation developed in this research provides a good basis for these purposes. In the next section, an example of how this empirical correlation can be used as a framework to design and optimize perfusive packings with respect to surface area and intraparticle convection will be discussed.

5.12 Optimization of Perfusive Packings

In the previous section, it has been found that the split ratio of intraparticle convection can be estimated for a bimodal PSD of perfusive packings by:

$$\frac{Q_p}{Q_b} = 1.516x10^8 (PI)^{3.67} (\frac{d_{macro}}{D_p})^{4.74} (\frac{d_{micro}}{D_p})^{0.21} (\frac{\sigma}{d})^{2.50}_{macro} (\frac{\sigma}{d})^{-1.51}_{micro} \qquad (5.26)$$

This split ratio, in terms of flow rates, can be rewritten into a split ratio of fluid linear velocities (Chapter 4):

$$K_u = \frac{u_p}{u_b} = 2.02x10^6 (PI)^{3.67} (\frac{d_{macro}}{D_p})^{4.74} (\frac{d_{micro}}{D_p})^{0.21} (\frac{\sigma}{d})^{2.50}_{macro} (\frac{\sigma}{d})^{-1.51}_{micro} \qquad ...(5.27)$$

where u_p is the intraparticle superficial velocity, and

u_b is the bed superficial velocty

The relevance of Equation (5.27) provides an *a priori* prediction of the intraparticle convective velocity (which is indicative of column efficiency) when the parameters characterizing the PSD of the packing are known. Furthermore, characterizing the PSD of the packing medium by mercury porosimetry is usually a routine analysis and hence plate height experiments are not required to elucidate the intraparticle flow.

The intraparticle Pe number, which is a measure of the significance of intraparticle convection over diffusion can then be obtained as:

$$Pe = \frac{u_p D_p}{\varepsilon_p D_e / \tau} \qquad ...(5.28)$$

Here, it is important to define the Pe number on a consistent basis for both the linear velocity and the effective diffusion coefficient; that is, both on an interstitial basis or both on a superficial basis. The difference between the two bases is given by the set of equations listed below:

$$D_e(\text{superficial}) = \varepsilon_p D_e(\text{interstitial}) \text{ and } u_p(\text{superficial}) = \varepsilon_p u_p(\text{interstitial}) \quad \text{...(5.29)}$$

The intraparticle Pe number is invariant with respect to the basis on which it is calculated as long as one maintains consistency. Since the intraparticle velocity is defined on a superficial particle basis, Equation (5.28) defines the intraparticle Pe number on this consistent basis:

$$Pe = \frac{\tau u_p D_p}{\varepsilon_p D_e}$$

ie. $$Pe = \frac{\tau K_u u_b D_p}{\varepsilon_p D_e} \quad \text{...(5.30)}$$

Therefore, for known values of $\varepsilon_p, D_e, D_p, \tau$ and u_b, Equation (5.27) for K_u provides a basis for calculating the intraparticle P_e number subject to changes in the parameters of the PSD and PI. An example will now be presented to demonstrate the use of the empirical correlation developed to design the porous structure of perfusive packing to meet specific demands of surface area and intraparticle convection.

POROS II packing will be used as the basis on which comparison will be made since POROS II has proved to be an improved medium compared to POROS I, For example, for POROS II, the adsorption capacity has been increased by at least 40% for lysozyme binding while maintaining the intraparticle convective transport behaviour for perfusion chromatography.

The tradeoffs between surface area and intraparticle convection can be illustrated by comparing the simulated results shown in Figure 5.37. Figure 5.37(a) plots the surface area of the macropores in the porous medium relative to that of POROS II on one ordinate and intraparticle Pe on another ordinate as a function of macropore diameter. Figure 5.37(b) plots the total surface area of the porous medium relative to the total surface area of POROS II on one ordinate and intraparticle Pe on another ordinate as a function of macropore diameter. Intraparticle Pe has been calculated using parameters of the PSD and PI of that of POROS II packing and Equation (5.30) at u_b=1000cm/h. The macroporous surface area of the packing was obtained from protein adsorption simulation (using lysozyme) as outlined in Chapter 4 for each of the lattice network generated using the PSD with varying macropore diameters. It has been found from mercury porosimetry that the macroporous surface area of the packing constitutes about 25-30% of the total surface area and the data in Figure 5.37(b) was therefore calculated on this basis. Other parameters used to obtain the results in Figure 5.37 are $\varepsilon_p = 0.61, D_e = 2.0x10^{-6} cm^2/s$ and $\tau = 6.0$ (Frey *et. al.*, 1993). In this study, the micropore size distributions were not changed as these have only marginal effects on intraparticle convection. Moreover, decreasing the mean micropore diameter to effect a higher adsorption capacity is inadvisable as hindered diffusion of protein solutes will be introduced at smaller pore diameters in addition to decreasing

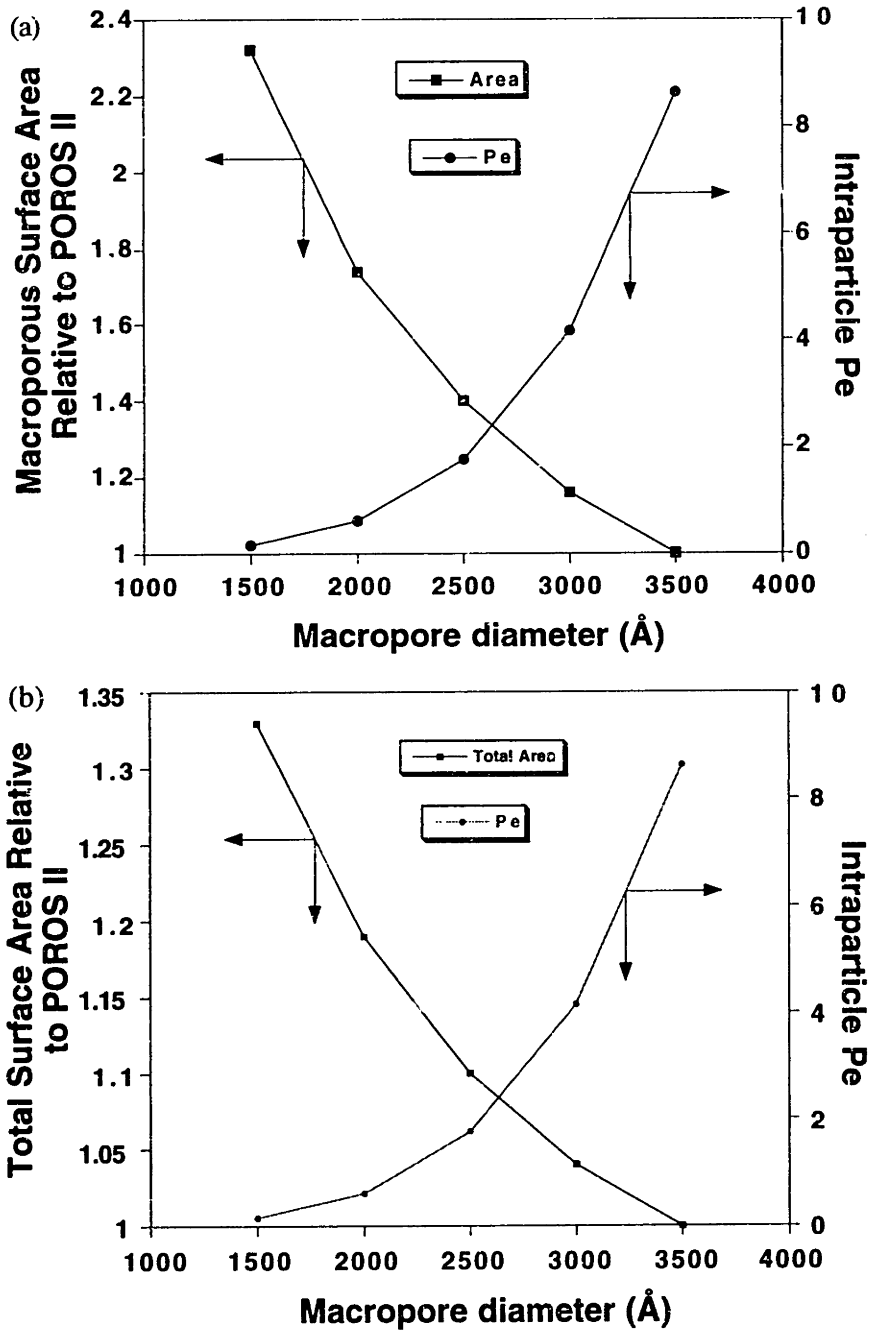


Figure 5.37 Illustration of the tradeoff between surface area and intraparticle Pe. The simulation data were obtained using parameters of the PSD and PI of POROS II. The linear velocity used for computing intraparticle convection is 1000 cm/h.

the intraparticle convective flow. The parameters of the micropore size distribution were deemed "optimum" in this study.

Based on the results in Figure 5.37(a), as the mean macropore diameter is decreased from 3500Å to 1500Å, the available macroporous surface area for adsorption increases by 2.3 times that of the POROS II base case. In terms of the total surface area (Figure 5.37(b)), decreasing the macropore diameter from 3500Å to 1500Å results in an increase in the total surface area by 1.33 times the total accessible surface area of POROS II. However, this is accompanied by a decrease in the intraparticle Pe from about 9 to almost zero at a bed velocity of 1000 cm/h. It is obvious that a compromise is required. Using the empirical correlation developed, there is a flexibility to vary two independent variables, the mean macropore size and the standard deviation of the macropore size distribution, to effect increased perfusion chromatography performance and this is illustrated below.

Shown in Figure 5.38 is a similar plot as Figure 5.37 but for two different standard deviations of the macropore size distribution. Two different perfusive packings can be made which have similar total surface area and amount of intraparticle convection. In this figure, for an intraparticle Pe number of 3.2 at 1000 cm/h, one packing can have a macropore diameter of 3000Å and a standard deviation of macropore size distribution of 1500Å while another can have a macropore diameter of 2800Å and a standard deviation of the macropore size distribution of 1600Å. Both of these have an approximate surface area of about 7% higher than that of POROS II.

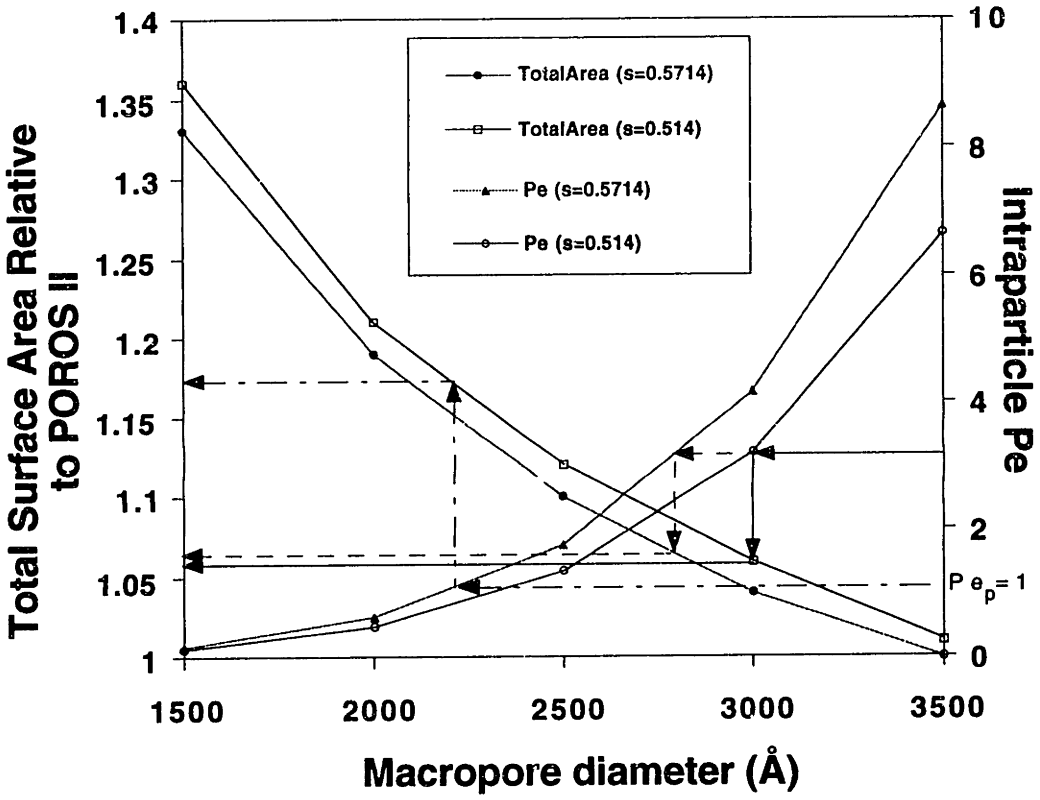


Figure 5.38 Illustration of the tradeoff between surface area and intraparticle Pe at different spreads of the macropore size distribution. The open symbols represent surface area and Pe curves of the same reduced standard deviation of the macropore size distribution, s=0.515, while the closed symbols are for s=0.571.

Unfortunately, Figure 5.38 also show that the improvement of the microstructure of perfusive packing with regards to surface area can only be improved marginally. In order to maintain intraparticle convection (intraparticle Peclet number, $Pe_p>1$ at a superficial velocity of 1000cm/hr), the size of the macropore diameter will have to be at least 2200Å and consequentlv, the maximum total surface area of the packing can only be increased by 18% more than that currently available in POROS II medium. Nonetheless, the empirical correlation developed can be useful for predicting the minimum size of the macropore diameter required to maintain intraparticle convection. This is illustrated by Figure 5.39.

Figure 5.39 shows the relationship between intraparticle Pe number as a function of macropore diameter at different superficial velocities. These simulation results reveal that for a fixed variance of the macropore distribution, as the macropore diameter is decreased, the critical superficial velocity for $Pe_p>1$ increases. This is because for smaller macropore diameters, the split ratio of intraparticle convection is consequently decreased and hence, a higher superficial velocity will be needed to maintain $Pe_p>1$. Based on the results in Figure 5.39, it can be seen that at a superficial bed velocity of 1000 cm/h, for example, the intraparticle Pe number is greater than 1.0 when the macropore diameter is larger than about 2200Å. This corresponds to a reduced macropore diameter of approximately $0.96x10^{-2}$. The minimum macropore diameter necessary for intraparticle convection, however, depends on the magnitude of the superficial bed velocity used in the chromatographic process. Frey and his co-workers (1993) have reported that the effects of intraparticle convection in chromatographic columns packed with large pore particles become significant when the reduced macropore diameter is greater than $1x10^{-2}$. Although their

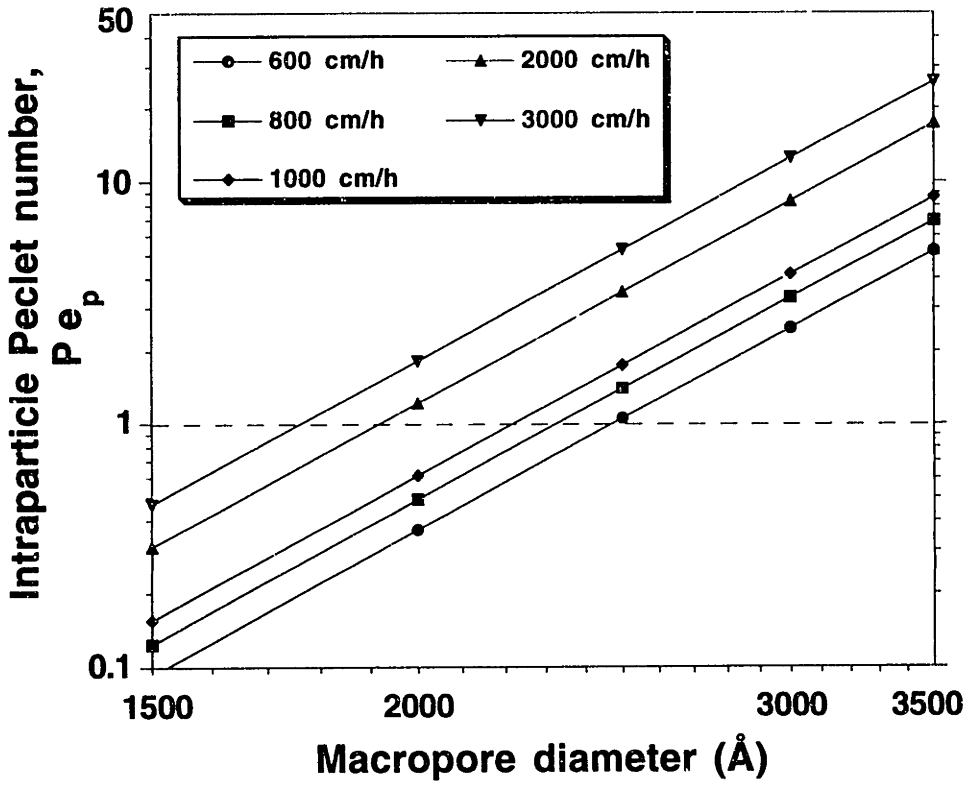


Figure 5.39 Relationship between intraparticle Pe and macropore diameter at different superficial velocities. The parameters of the PSD and PI used are those of POROS II packing.

claim of this minimum reduced macropore diameter is consistent with that calculated using the empirical correlation at 1000 cm/h, this was neither theoretically nor experimentally supported. On the other hand, the results in Figure 5.39 can be used to provide the necessary conditions to define the minimum macropore diameter required for intraparticle convection which is dependent on the superficial bed velocity.

In conclusion, in this section, it has been shown from the results derived from the empirical correlation on the microstructure of perfusive packing can be tailored (albeit only marginally) with respect to meeting different specifications of total surface area and intraparticle convection by adjusting the mean macropore size and the standard deviation of the macropore size distribution. In addition, the empirical correlation developed can be used to specify the minimum macropore diameter required for intraparticle convection in perfusive packings depending on the superficial bed velocity used in the chromatographic process.

5.13 Limitations of Network Model

To conclude this chapter, it seems necessary to present the limitations of the network model developed:

(a) The fixed lattice size of 25x25x25 of the network can only be used to represent perfusive particles smaller than 20μm nominal particle diameter. To model larger particles, a much larger lattice will be necessary. This, however, does not change the modelling techniques, nor the empirical correlation for intraparticle Pe. The correlation has been developed in terms of dimensionless parameters of the PSD. Larger lattices will be required if the PSD and PI of larger particles have to be elucidated by the mercury porosimetry simulation technique. The only strain here is the use of more computer memory and the necessity of longer computation time.

(b) The PI has been assumed to be a constant for the HPLC column. In actual fact, the pore interconnectivity of the intraparticle pores may be different from the pore interconnectivity of the interstitial pores. In this case, varying PI's should be used for generating the lattice network. Fortunately, this is not a strong restriction of the model as incorporation of varying PI's is rather straightforward since the assignment of the interstitial pores and the intraparticle pores was undertaken independently. Moreover, the PI of porous structures is difficult to obtain (Portsmouth and Gladden, 1992) and is usually treated as a fitted parameter. As a result, it is sufficient that the PI can be taken to be an overall constant for the HPLC column.

(c) The empirical correlation developed for the intraparticle Pe number is only valid for bed porosity of about 25% and particle porosity of about 60%. A different proportionality constant is anticipated for other values of bed and particle porosities. However, the proportionality constant may be a complex function of porosity, as Dullien (1992) has pointed out and may almost be impossible to elucidate. Fortunately, for most closely packed HPLC packings, the bed porosity does not differ too much from 25-30% and the porosity of the packing medium is usually in the range around 60%. As such, the correlation can be used to approximate intraparticle convection behaviour for most HPLC packings.

(d) The empirical correlation has been developed for a PSD that is bimodal. For packings which have unimodal PSD, a different correlation will have to be elucidated. However, in order to take advantage of intraparticle convection in perfusion chromatography, large-pore packing media, like perfusive media, are always made with bimodal PSD rather unimodal PSD.

Chapter 6 -

Conclusions

This thesis presented work on establishing a discrete pore network model to represent a perfusion chromatography column. The areas covered in the research included development of the network model, validation of the model with regards to static experiments of mercury porosimetry of blended perfusive media, protein adsorption and size exclusion chromatography of polystyrene molecules and dynamic experiments of permeability and column efficiency. Application of the network model for optimization studies was also investigated. A summary of the conclusions from this thesis is outlined as:

(1) A lattice network model was established for a representative section of the packed column and mercury intrusion was simulated using the model. Using scanning electron microscopy (SEM) and digital image analysis, it was found that the interstitial and macropore size distribution can be represented by the Gaussian distribution. As for the micropores, sensitivity analysis revealed that the Gaussian distribution is also a good representation of the micropore size distribution. The quantitative values

of the input parameters (PSD and PI of the packing) for the model were obtained from matching the simulated mercury intrusion data to experimentally obtained mercury porosimetry curves. Sensitivity analyses were carried out to optimize the lattice size and to ascertain the uniqueness of the input parameters obtained. The results indicate that the PSD of POROS packings comprises of a group of macropores (1000's Å range) and a group of micropores (100's Å range) which are consistent with measurements inferred from SEM. The model established provides a useful tool for obtaining the PSD of porous materials when other methods like mercury porosimetry involve erroneous data interpretation and SEM and densitometry measurements are too laborius. Using the model, the pore number distribution was discretized on a per particle basis which provides a simple semi-quantitative comparison between different perfusive materials with regards to protein adsorption capability and through-put.

(2) Mercury intrusion was simulated using the network model for blends of POROS I and II. The excellent agreement between simulation and experimental data not only corroborated with the network model established, but it also confirms the PSD parameters obtained for POROS I and II. More importantly, the results indicate that the network model can be used as a quality control/assurance tool.

(3) By simulating the adsorption capacities for proteins of varying sizes on POROS I and II materials, the network model was used to study the accessible porous surface area available for adsorption. The agreement between experimental data and model predictions were excellent. The

results also demonstrate the importance of pore interconnectivity on the accessibility of the porous structure by a comparison of the accessible surface area for the network model against a parallel pore model whereby the pores are not connected to each other. It has also been shown from the simulations that there is no competitive advantage of POROS II over POROS I, in terms of adsorption capacity, when the solute molecule exceeds a critical size. In this case, accessibility to the pores in POROS II can be decreased due to interconnectivity between the pores when pores which provide the surface area are shielded by smaller pores nearer the surface.

(4) Another independent verification of the model was achieved by simulating size exclusion chromatography (SEC). Experimentally, SEC was performed using polystyrene standards of small polydispersities. Principles of statistical mechanics and Monte Carlo techniques were employed to provide the framework for steric exclusion of the flexible macromolecules in the pores. The agreement between simulation and experimental data indicates that the model can be used to simulate SEC, in addition to independently validating the use of the model for subsequent studies of transport mechanism in perfusive packings.

(5) The dependence of column efficiency, HETP, on the pore structure of the packings was demonstrated by residence time distribution studies on columns packed with POROS I and II and blends of the pure materials; the mixtures of POROS I and II were used to obtain varying overall PSD present in the packed columns. Experimental results reveal that the particle porosity alone cannot characterize the packing and that the PSD

should be used. It was further demonstrated that the efficiency of the column can be increased by using packings with larger pores. For perfusive packings which allow a lower intraparticle velocity, the HETP results demonstrate that a higher superficial velocity is necessary to trigger intraparticle convection. The minimum superficial velocity for the onset of intraparticle Pe=1 is therefore reckoned as an important consideration for optimizing the use of perfusive packings, apart from adsorption capacity.

(6) Permeability of the perfusive medium was studied by incorporating a transport methodology into the network model. The split ratios of flow rates and superficial velocities in the particle to those in the column were elucidated for packed beds of POROS I, II and blends of the pure packings. The results were confirmed by a comparison with existing macroscopic models. The excellent agreement between the split ratios predicted by the network model and the macroscopic models provide a strong validation of the transport methodology used.

(7) Simulations to probe the effects of changes in parameters describing the PSD on the split ratio were performed using the validated transport methodology in the network model. Parameters of the PSD and PI were varied and the effects of such variations on the magnitude of intraparticle convection were elucidated. Based on the study, an empirical correlation which relates intraparticle convection to the microscopic parameters of the PSD and PI was developed. It was found that intraparticle convection is a strong function of PI and the parameters of the macropore size distribution. On the other hand, it was concluded that intraparticle

convection is weakly dependent on the mean sizes of the micropores and that intraparticle convection is inversely proportional to the spread of the micropore size distribution at constant bed and particle porosities. Furthermore, the simulations indicate that the Blake-Kozeny equation may not be appropriate for quantifying intraparticle permeability accurately. Experimental data reveal that intraparticle permeability may vary as pore diameter raised to an exponent of 2.58-2.73 as opposed to an exponent of 2 as indicated by the Blake-Kozeny equation. Using the empirical correlation developed and knowing the parameters which characterize the PSD of the packing, it is possible to predict *a priori*, the intraparticle convective velocity.

(8) An optimization study was attempted using the empirical correlation developed. Tradeoffs between adsorption capacity and throughput were illustrated with the conclusion that column efficiency has to be sacrificed for capacity and vice-versa. Based on the results of the network model, it was found that there is a flexibility of having two independent variables which can be varied for optimization purposes - the mean macropore size and the standard deviation of the macropore size distribution. An optimized design of the porous structure of perfusive packings can only be determined depending on the desired capacity and/or column efficiency.

Chapter 7 -

Recommendations for Future Research

The development of a fully validated network model for characterizing the PSD of perfusive media has been presented in this thesis. The work presented suggests several areas for continuing research on using the network model to "tailor" perfusion chromatography to specific demands. The areas which are discussed in this section concern automation of the intrusion simulation technique, further validation of the empirical correlation for predicting intraparticle convection and correlating polymerization technique and reaction conditions to parameters of the PSD.

In this thesis, the parameters of the PSD and PI of the perfusive media were inferred by a trial and error approach of matching simulated intrusion data

to experimentally derived intrusion curves. The objective function used is based on the goodness-of-fit factor of the match between the two sets of data. The input parameters for generating the network model were varied manually (albeit systematically) and the effect of the variations on the goodness-of-fit factor was used to refine the parameters. However, it has been shown (Himmelblau, 1972) that a trial and error method may not be practical if more than three or four variables are involved. This technique can certainly afford to be more efficient by incorporating a numerical optimization of the non-linear multivariable problem. Efficiency is important here since the problem requires an iterative solution procedure.

Different non-linear multivariable optimization methods can be used for this purpose (Edgar and Himmelblau, 1988) - direct and indirect methods. Among the direct methods, which do not require information about derivative values of the objective function in determining the search direction, the method of "sequential simplex" formulated by Spendley and co-workers (1962) is recommended. Indirect methods, however, have been demonstrated to be more efficient than direct methods. Among indirect methods, high-order finite difference techniques may be applicable. The choice of a particular technique for computerizing the mercury intrusion simulation depends on the optimization of a number of criteria: the robustness or success in obtaining an optimal solution to within a specified precision, the number of equivalent function evaluations and the most important of all, the central processing time to termination to within the desired degree of precision. With the incorporation of a computerized numerical optimization technique, data interpretation of mercury porosimetry using the network model will be more efficient, in addition to being accurate.

Additional validation of the empirical correlation for predicting intraparticle convection is another area for continuing research. Although the empirical correlation developed is valid for perfusive media of, basically, any nominal particle diameters since the parameters involved in the relationship are dimensionless with respect to the particle diameter, the simulations were validated with experimental data based only on particle diameters of 20μm. Further validation of the correlation is recommended by using nominal 10μm and 50μm perfusive particles and possibly blends of POROS I and POROS II of each of the media. Plate height analyses may be performed for packed columns of the abovementioned packings and using the modified van Deemter equation of Rodrigues *et. al.*, (1991), the intraparticle superficial velocity can be calculated. These can then be compared with the prediction of the empirical correlation for more validation. In addition to using particles of various diameters, perfusive media of different PSD can also be used for corroboration of the intraparticle convection correlation. The final area for future research is therefore in the production of perfusive media with various PSD.

Perfusive media have been manufactured by advanced techniques in suspension polymerization. The conditions and procedure of the polymerization is very important as it controls the ultimate properties of the final polymer. Frechet (1993) and Sherrington (1993) have demonstrated that the use of an appropriate porogenic solvent, like toluene, can be used to vary the PSD of the resultant resins. By simple adjustments of the composition, size and molecular weight of the porogenic polymer, polystyrene-divinylbenzene resins of significantly different PSD have been produced. It is recommended that different polymerization conditions and methods be tried to manufacture perfusive media

of different PSD and possibly PI. These polymerization conditions can then be correlated to the correct PSD using results of the network model since the network model established in this thesis can accurately characterize the PSD of the perfusive media. Subsequently, different porous structures of perfusive media can be produced and optimization results can then be practically realized.

Nomenclature

Abbreviations and Symbols:

A(r) = accessible area of solute radius, r

a,b = parameters of two-parameter log-normal distribution

$\bar{D}$ = mean diameter for skewness factor calculation

D = solute diameter (SEC) or pore diameter (PSD)

D_{ax} = axial dispersion coefficient of bed

D_e = pore diffusivity = solute bulk diffusivity corrected for finite size of solute

D_{eff} = pore diffusivity of solute in macropores

D_i = particle (or macropore) diameter for skewness factor calculation

D_m = solute bulk diffusivity

d_{macro} = mean macropore diameter

d_{micro} = mean micropore diameter

D_p = particle diameter

d_p = pore diameter (Chapter 2)

E = exclusion probability

F = goodness-of-fit factor

$f(\lambda)$ = intraparticle convection function defined by Rodrigues *et.al.* (1982)

F(R) = accessible volume fraction of pores in network taken up in radii R

F_i = probability plot position

HETP = height equivalent to a theoretical plate

HPLC = high performance (pressure) liquid chromatography

K = exclusion coefficient

K_o = partition coefficient based on Casassa (1972)

K_p = affinity constant of protein for protein surface

K_{P-S} = bed permeability based on analysis of Park and Stephanopoulos (1993)

K_u = split ratio of superficial velocities

L = length of column (Chapter 3) or lattice size (Chapter 4)

l = segment length of macromolecule

L_e = entrance length

l_p = pore length

M = relative molecular weight of polystyrene molecule

N = number of transfer units

n = number of segments of macromolecule

n = total number of particles (pores) counted (SEM)

n_t = number of transfer units for diffusion in throughpores

p_c = percolation threshold

Pe = Peclet number

Pe_b = bed Peclet number

PI = pore interconnectivity

p_i = relative mathematical proportion of pores of class i

P_o = equilibrium pressure in capillary condensation or equilibrium pressure in mercury porosimetry

PSD = pore size distribution

Q = pore flow rate

q = equilibruim protein adsorbed

Q_b = total flow rate through the bed

Q_{inst} = total liquid flow through interstitial pores

q_{max} = maximum static capacity

Q_p = intraparticle flow rate

Q_{pore} = total liquid flow through intraparticle pores

Q_{red} = reduced split ratio of flow rates defined by Eqn (5.25)

Q_{total} = total liquid flow through bed

R = pore radius

r = solute radius

Re = Reynolds number

r_g = radius of gyration of macromolecule

$R_{h,pore}$ = hydraulic radius of pores

RTD = residence time distribution

S = column cross-sectional area

s = reduced standard deviation of macropore size distribution

SEC = size exclusion chromatography

SEM = scanning electron microscopy

S_{eq} = equilibrium protein concentration

S_k = skewness factor

TEM = transmission electron microscopy

u_b = superficial bed velocity

u_p = intraparticle superficial velocity

V = cumulative intrusion volume

V_c = column volume

V_e = elution volume

V_i = volume of intraparticle pores

V_{int} = specific volume of mercury intruded into particles

V_m = total volume of voids in column

V_o = volume of interstitial pores

W = peak width at half peak height

y_{iexpt} = experimental pressure at intrusion volume V_i

y_{isim} = pressure at simulated intrusion volume V_i

Z = coordination number of lattice

Greek Symbols:

α_1 = first zero of Bessel function, $J_0(x) = 0$

ΔP = pressure drop across column

ε_c = column porosity

ε_p = particle porosity

ε_t = total porosity of column

γ = surface tension of mercury

η = ratio of solute diameter to macropore diameter

λ = intraparticle Pe defined by Rodrigues *et.al.* (1982)

λ_G = ratio of radius of gyration of macromolecule to pore radius

μ_i = mean diameter of pores of class i

μ_1 = mean residence time of eluite

μ = liquid viscosity

ρ_l = liquid density

ρ = skeletal density of bead particles

σ_i = standard deviation of diameter of pores of class i

σ_b^2 = variance of velocity distribution over bed cross-section

σ = standard deviation of eluite peak

θ = contact angle of mercury with solid

τ = tortuosity factor

References

Abbasi, M.H., Evans, J.W. and Abramson, I.S., Diffusion of Gases in Porous Solids: Monte Carlo Simulations in the Knudsen and Ordinary Diffusion Regime, *A.I.ChE. J.*, **29**, 617, (1983)

Afeyan, N.B., Fulton, S.P. and Regnier, F.E., Perfusion Chromatography Packing Materials for Proteins and Peptides, *J. Chromatogr.*, **544**, 267, (1991).

Afeyan, N.B., Fulton, S.P., Gordon, N.F., Mazsaroff, I., Varady, L. and Regnier F.E., " Perfusion Chromatography: An Approach to Purifying Biomolecules" *Biotechnol*, **8**, 203, (1990a).

Afeyan, N.B., Gordon, N.F., Mazsaroff, I., Varady, L., Fulton, S.P., Yang, Y.B. and Regnier, F.E., Flow-through Particles for the HPLC Separation of Biomolecules: Perfusion Chromatography, *J. Chromatogr.*, **519**, 1, (1990b).

Akanni, K.A., Evans, J.W. and Abranson, I.S., Effective Transport Coefficients in Heterogeneous Media, *Chem. Eng. Sci.*, **42** (8), 1945, (1987).

Akima, H., A New Method of Interpolation and Smooth Curve Fitting based on Local Procedures, *J. ACM*, **17**, 589, (1970)

Androutsoupoulos, G.P. and Mann, R., On the Inevitability of Nonuniform Foulent Deposition within a Catalyst pellet, *Chem. Eng. Sci.*, **33**, 673, (1978).

Applegate, M., Development and Characterization of Macroporous Ceramic Matrix Bioreactors for Mammalian Cell Culture, Ph.D Thesis, M.I.T., (1991)

Arnold, F.H., Chalmers. J.J., Saunders, M.S., Croughan, M.S., Blanch, H.W. and Wilke, C.R., A Rational Approach to the Make-up of Affinity Chromatography, in "Purification of Fermentation Products", D. LeRoith Ed., *Amer. Chem. Soc. Symp. Ser.*, **211**, 113, (1985).

Atkinson, B., Brocklebank, M.P., Card, C.C. and Smith, J.M., Low Reynolds Number Developing Flow, *A.I.ChE.J.*, **15**, 548, (1969)

Barett, E.P., Joyner, L.G. and Halenda, P.P., The Determination of Pore Volume and Area Distributions in Porous Substances. I. Computations from Nitrogen Isotherms *J. Am. Chem. Soc.*, **73**, 373, (1951)

Barett, K.E.J., in "Dispersion Polymerization in Organic Media", Wiley, NY, (1975)

Batia, B.J., Ph.D Thesis, Univ. of Waterloo, Waterloo, Ontario, (1973)

Bell., W.K., Brakel, J. Van and Heertjes, Mercury Penetration and Retraction Hysteresis in Closely Packed Spheres, P.M., *Powder Tech.*, **29**, 75, (1981)

Benzoni, J. and Chang, H.-C., Effective Diffusion in Bi-disperse Media - An Effective Medium Approach, *Chem. Eng. Sci.*, **39**, 161, (1984)

Berman, D., Orr, B.G., Jaeger, H.M. and Goldman, A.M., Conductance of Filled Two-Dimensional Networks, *Phys. Rev. B*, **33**, 4301, (1986)

Bernardini, F., Collepardi, M. and Armisi, I., *Chim. Ind. (Milan)*, **49**, 366, (1967)

Brandt, W.W., Use of Percolation Theory to Estimate Effective Diffusion Coefficients of Particles Migrating on Various Ordered Lattices and in a Random Network Structure, *J. Chem. Phys.*, **63**, 5162, (1975)

Brenner, H. and Gaydos, L.J., The Constrained Brownian Movement of Spherical Particles in Cylindrical Pores of Comparable Radius, *J. Colloid. Interface Sci.*, **58**, 312, (1977)

Broadbent, S.R., and Hammersley, J.M., Percolation Processes. I. Crystals and Mazes, *Proc. Camb. Phil. Soc.*, **53,** 629, (1957).

Brunauer, S., Emmett, P.H. and Teller, E., Adsorption of Gases in Multimolecular Layers, *J. Am. Chem. Soc.*, **60**, 309, (1938)

Brunauer, S., Mikhail, R. SH. and Bodor, E. F., Pore Structure Analysis without a Pore Shape Model, *J. Colloid Interface Sci.*, **24**, 451, (1967)

Burganos, V.N. and Sotirchos, S.V., Diffusion in Pore Networks: Effective Medium Theory and Smooth Field Approximations, *A.I.ChE. J.*, **33**, 1678, (1987)

Cardy, J.L., Field Theoretic Formulation of an Epidermic Process with Immunization , *J. Phys. A*, **16**, L709, (1983)

Carta, G. and Rodrigues, A.E., Diffusion and Convection in Chromatographic Processes using Permeable Supports with a Bi-disperse Pore Structure, *Chem. Eng. Sci.*, **48**, 3927, (1993)

Carta, G., Gregory, M.E., Kirwan, D.J. and Massaldi, H.A., Chromatography with Permeable Supports: Theory and Comparison with Experiments, *Sep. Tech.*, **2**, 62, (1992)

Chang, S.H., Gooding, K.M. and Regnier, F.E., High-Performance Liquid Chromatography of Proteins, *J. Chromatogr.*, **125**, 103, (1976)

Cheng, C.M., Micale, F.J., Vanderhoff, J.W.and El-Aasser, M.S., Synthesis and Characterization of Monodisperse Porous Polymer Particles, *J. Polym. Sci, Polym. Chem.* **30**, 235, (1992a)

Cheng, C.M., Vanderhoff, J.W. and El-Aasser, M.S., Monodisperse Porous Polymer Particles: Formation of the Porous Structure, *J. Polym. Sci, Polym. Chem.*, **30**, 245, (1992b)

Chiantore, O. and Guaita, M., "High Performance SEC with Microparticulate Porous Silica Spheres: Influence of Flow Rate, Solute Mass Transfer and Polydispersity of Samples", *J. Liq. Chromatogr.*, **5** (4), 343 (1982).

Colin, H. and Crichon, G., Development and Use of Carbon Adsorbents in High-Performance Liquid-Solid Chromatography, I. Carbon-Coated Silica Particles, *J. Chromatogr.*, **126**, 43, (1976)

Conner, W.C. and Lane, A.M., Measurement of the Morphology of High Surface Area Solids: Effect of Network Structure on the Simulation of Porosimetry, *J. Catalaysis*, **89**, 217, (1984).

Coupek, J., Krivakova, M. and Pokorny, S., *J. Polym. Sci., Polym. Symp.*, **42**, 185, (1973)

Cox, M.A.A., Mercury Injection Measurements used in the Prediction of Rock Pore Dimensions Employing a Crystalline Lattice of Capillaries, *Chem. Eng. Sci.*, **46**, 167, (1991)

Cranston, R.W. and Inkley, F.A., Determination of Pore Structures from Nitrogen Adsorption Isotherms, *Adv. Catalysis*, **9**, 143, (1957)

D'Agostino, R.B., An Omnibus Test of Normality for Moderate and Large Sample Size, *Biometrika*, **58**, 341, (1971)

Daccord, G., Chemical Dissolution of a Porous Medium by a Reactive Fluid, *Phys. Rev. Lett.*, **58**, 479, (1987)

Dagan, Z., Weinbaum, S. and Pfeffer, R., An Infinite-Series Solution for the Creeping Motion Through an Orifice of Finite Length, *J. Fluid Mech.*, **115**, 505, (1982)

de Gennes, P.G., La Percolation: Un Concept Unification, *La Recherche*, **7**, 919, (1976)

Deen, W.M., Hindered Transport of Large Molecules in Liquid-Filled Pores, *A.I.ChE. J.*, **33** (9), 1409, (1987).

Deen, W.M., Personal Comm., (1994)

Derrida, B., Stauffer, D., Herrmann, H.J. and Vannimenus, J., Transfer Matrix Calculation of Conductivity in the Three-Dimensional Random Resistor Networks at Percolation Threshold, *J. Phys. Lett.*, **44**, L701, (1983)

Dhawan, G.K., Photomicrographic Investigation of the Structure of Sandstones and other Porous Media by the Methods of Quantitative Stereology, PhD Thesis, University of Waterloo, Waterloo, Ontario, (1972)

Domb, C. and Green, M.S., Eds., in "Phase Transitions and Critical Phenomena", Vols. 1-13, Academic Press, NY, (1972-1989)

Dullien, F.A.L., Single Phase Flow through Porous Media and Pore Structure, *The Chem. Eng. J.*, **10**, 1, (1975)

Dullien, F.A.L., New Network Permeability Model of Porous Media, *A.I.ChE. J.*, **21**, 299, (1975a)

Dullien, F.A.L., Single Phase Flow through Porous Media and Pore Structure, *Chem. Eng. J.*, **10**, 1, (1975b)

Dullien, F.A.L., Wood's Metal Porosimetry and its Relation to Mercury Porosimetry, *Powder Tech.*, **29**, 109, (1981)

Edgar, T.F. and Himmelblau, D.M., in "Optimization of Chemical Processes", McGraw Hill, NY, (1988)

Elam, W.T., Kerstein. A.R. and Rehr, J.J., Critical Properties of the Void Percolation Problems for Spheres, *Phys. Rev. Lett.*, **52**, 1516, (1984)

Engelhardt, H. and Mathes, D., Chemically Bonded Stationary Phases for Aqueous High-Performance Exclusion Chromatography, *J. Chromatogr.*, **142**, 311, (1977)

Evans, J.W., Abbasi, M.H. and Sarin, A., A Monte Carlo Simulation of the Diffusion of Gases in Porous Solids, *J. Chem. Phys.*, **72** (5), 2967, (1980).

Evans, M., Palmer, D.A., Miller, J.N. and French, M.T., Flow Injection Fluorescence Immunoassay for Serum Phenytoin using Perfusion Chromatography, *Anal. Proc.*, **31**, 7, (1994)

Fatt, I., The Network Model of Porous Media, *Petrol Trans. AIME*, **207**, 144, (1956).

Feng, C., Kostrovs. V.V. and Stewart, W.E., Multicomponent Diffusion of Gases in Porous Solids: Models and Experiments, *Ind. Eng. Chem. Fundam.*, **12**, 143, (1973).

Fisher, M.E., in "Critical Phenomena: Enrico Fermi Summer School", M.S. Green, Ed., Academic Press, NY, (1971)

Flory, P.J., Molecular Size Distribution in Three Dimensional Polymers, *J. Am. Chem. Soc.*, **63**, 3083, (1941).

Frechet, J.M.J., Design and Preparation of Novel Particulate and Continuous Polymeric Macroporous Media for the Separation of Biological and Synthetic Molecules, *Makromol. Chem., Macromol. Symp.*, **70/71**, 289, (1993)

Freeman, D.H. and Poinescu, I.C., Particle Porosimetry by Inverse Gel Permeation Chromatography, *Anal. Chem.*, **49**, 1183, (1977)

Frey, D., Schweinheim, E. and Horvath, CS., The Effect of Intraparticle Convection on the Chromatography of Biomolecules, *Biotech. Prog.*, **9**, 273, (1993)

Fulton, S.P., Meys, M., Varady, L., Jansen, R. and Afeyan, N.B., Antibody Quantitation in Seconds using Affinity Perfusion Chromatography, *Biotech.*, **11**, 226, (1991)

Giddings, J.C., "Dynamics of Chromatography: Part I. Principles and Theory", Marcel Dekker, New York, (1965).

Giddings, J.C., Kucera, E., Russel, C. and Meyers, M., Statistical Theory for the Equilibrium Distribution of Rigid Molecules in Inert Porous Networks. Exclusion Chromatography. *J. Phys. Chem.*, **78**, 397, (1968)

Gladden, L.F., Influence of Pellet Structure on Selectivity During Swing Adsorption Separations, *Chem. Eng. Sci.*, **46**, 2455, (1991)

Graham, E.E., Chen F.F., Rate of Protein Absorption and Desorption on Cellulose Ion Exchanger, *A.I.ChE. J..*, **28**, 245, (1982).

Grimaud, E., Lecoq, J.C., Boschetti, E. and Corgier, M., Comparison of Gels used for Molecular Sieving of Proteins by Electron Microscopy and Pore Parameters Determinations, *J. Chromatogr.*, **166**, 37, (1978)

Guerra, V.O., Tapia, R.A. and Thompson, J.R., A Random Number Generator Function Continuous Random Variables based on an Interpolation Procedure of Akima, *in Proceedings of the Ninth Interface Symposium on Computer Science and Statistics*, B. Hoaglin and Welsh, Eds., Prindle, Weber and Schmidt, Boston, 228, (1976)

Gunn, D.J., Axial Dispersion in Pacjed Beds : The Effect of the Quality of Packing, *Trans. Instn. Chem. Engrs.*, **49**, 109, (1971)

Halasz, I. and Martin, K., Pore Sizes of Solid, *Angew Chem. Int. Ed. Engl.*, **17**, 901, (1978)

Haller, W., Rearrangement Kinetics of the Liquid-Liquid Immiscible Microphases in Alkali Borosilicate Melts, *J. Chem. Phys.*, **42**, 686, (1965)

Hashimoto, K., and Silveston, P.L., Gasification : Part I. Isothermal Kinetic Control Model for a Solid with a Pore Size Distribution, *A.I.ChE. J.*, **19,** 259, (1973).

Hashimoto, T., Macroporous Synthetic Hydrophilic Resin-Based Packings for the Separation of Biopolymers, *J. Chromatogr.*, **544**, 249, (1991a)

Hashimoto, T., Non-porous Hydrophilic Resin-Based Packings for the Separation of Biopolymers, *J. Chromatogr.*, **544**, 257, (1991b)

Hashimoto, T., Sesaki, H., Auira, M. and Kato, Y., *J. Polym. Sci., Polym Phys. Ed.*, **16**, 1789, (1978)

Haughey, D.P. and Beveridge, G.S.G., Structural Properties of Packed Beds - A Review, *Can. J. Chem. Eng.*, **47**, 130, (1969)

Hegedus, R.D., The Dependence of Performance on fiber Uniformity in Aligned fiber HPLC columns, *J. Chromatogr. Sci.*, **26**, 425, (1988)

Heiba, A.A., Sahimi, M., Scriven, L.E., and Davis, H.T., Percolation Theory of Two-Phase Relative Permeability, SPE 11015, New Orleans, L.A., (1982).

Himmelblau, D.M., in "Applied Nonlinear Programming", McGraw Hill, NY, (1972)

Hjerten, S., Liao, J.-I. and Zhang, R., High Performance Liquid Chromatography on Continuous Polymer Beds, *J. Chromatogr.*, **473**, 273, (1989)

Hjerten, S. and Mosbach, R., "Molecular-Sieve" Chromatography of Proteins on Columns of Cross-Linked Polyacryamide, *Anal. Biochem.*, **3**, 109, (1962)

Hjerten, S., The Preparation of Agarose Spheres for Chromatography of Molecules and Particles, *Biochem. Biophys. Acta*, **79**, 393, (1964)

Hollewand, M.P. and Gladden, L.F., Modeliing of Diffusion and Reaction in Porous Catalysts using a Random Three-Dimensional Network Model., *Chem. Eng. Sci.*, **47**, 1761, (1992)

Holm, E.A., A Percolation Model for the Zero-Shrinkage Whisker Fraction in Ceramic Matrix - Ceramic Whisker Composites, M.Sc. Thesis, MIT, (1989).

Holshen, J. and Kopelman, R., "Percolation and Cluster Distribution I. Cluster Multiple Labeling Technique and Critical Concentration Algoritm", *Phys. Rev. B*, **14**, 3438, (1976).

Horak, Z. and Schneider, P., " Comparison of some Models of Porous Media for Gas Diffusion", *Chem. Eng. J.*, **2**. 26, (1971).

Horsfield, H.T., The Strength of Asphalt Mixtures, *J. Soc. Chem. Ind.*, **53**, 107T, (1934)

Imdakm, A.O. and Sahimi, M., Computer Simulation of Particle Transport Processes in Flow Through Porous Media, *Chem. Eng. Sci.*, **46**, 1977, (1991)

Imdakm, A.O. and Sahimi, M., Transport of Large Particles in Flow Through Porous Media, *Phys. Rev. A*, **36**, 5304, (1987)

IMSL, in "Fortran Subroutines for Statistical Analysis", IMSL, Inc., Houston, Texas, (1991)

Jerauld, G.R., Hatfiled, J.C., Scriven, L.E. and Davis, H.T., Percolation and Conduction on Voronoi and Triangular Networks: A Case Study in Topological Disorder, *J. Phys.* **C17**, 1519 (1984).

Kalghatgi, K. and Horvath, CS., Rapid Peptide Mapping by HPLC, *J. Chromatogr.* **443**, 343, (1988).

Kalghatgi, K., Personal Comm., (1994)

Kerker, M., in "The Scattering of Light and Other Electromagnetic Radiation", Acad. Press, NY, (1969)

Kerstein, A.R. and Bug, A.L.R., Scaling Theory of Pore Growth in a Reactive Solid, *Phys. Rev. B.*, **34,** 1754, (1986)

Kerstein, A.R., Equivalence of the Void Percolation Problem to Overlapping Spheres and a Network Problem, *J. Phys. A*, **16**, 3071, (1983)

Kim, D.H. and Johnson, A.F., Computer Model for GPC of Polymer, in "Size Exclusion Chromatography: Methodology and Characterization of Polymers and Related Materials", *ACS Symp. Series*, **245**, 25 (1984).

Kreith, F. and Eisenstandt, R., Pressure Drop and Flow Characteristics of Start Capillary Tubes at Low Reynolds Numbers, *Trans. ASME*, **79**, 1070, (1957)

Koplik, J. and Lasseter, T.J., One- and Two-Phase Flow in Network Models of Porous Media, *Chem. Eng. Commun.*, **26,** 285, (1984)

Kreveld, V. M.E. and Van den Hoed, N., Mass Transfer Phenomena in GPC, *J. Chromatogr.*, **149**, 71 (1978).

Kuga, S., Pore Size Distribution Analysis of Gel Substances by Size Exclusion Chromatography, *J. Chromatogr.*, **206**, 449, (1981)

Lane, A., Shah, N. and Conner, W.C., Measurement of the Morphology of High-Surface Area Solids: Porosimetry as a Percolation Problem, *J. Colloid Interface Sci.*, **69**, 235, (1986)

Lane, A.M., A Network Model for Interpretation of Mercury Porosimetry Data, Ph.D. Thesis, University of Massachusetts, (1983).

Lapidus, G.R., Lane, A.M., Ng, K.M. and Conner, W.C., Interpretation of Mercury Porosimetry Data using a Pore-Throat Network Model, *Chem. Eng. Commun.*, **38**, 33, (1985)

Lehman, E.D., Joyce, J.G., Freymeyer, D.K. II, Bailey, F.J., Herber, W.K. and Miller, W. J., A Novel Process for the Large-Scale Purification of Recombinant Tick Anticoagulant Peptide using Perfusion Chromatography, *Biotech.*, **11**, 207, (1993)

Levenspiel, O., in "Chemical Reaction Engineering", 2nd Ed., John Wiley & Sons, NY, (1972)

Liapis, A.I. and McCoy, M.A.. A Theory of Perfusion Chromatography, *J. Chromatogr.*, **599**, 87, (1992)

Lloyd, L. and Warner, F.P., Preparative HPLC on a Unique High-speed Macroporous resin, *J. Chromatogr.*, **512**, 365, (1990)

Lu, Z.P., Loureiro, J.M. and Rodrigues, A.E., Single-Pellet Cell for the Measurement of Intraparticle Diffusion and Convection, *A.I.ChE. J.*, **38**, 416, (1992)

MacKay, G. and Jan, N., Forest Fires as Critical Phenomena, *J. Phys. A*, **17**, L757, (1984)

Majors, R.E., Polymeric Liquid Chromatographic Column Technology in Japan, *LC-GC*, **11**, 778, (1993)

Mann, R. and Golshan, H., Application of a Stochastic Network of Pore Model to a Catalyst Pellet, *Chem. Eng. Commun.*, **12**, 377, (1981).

Mann, R., Almeida, J.J. and Mugerwa, M.N., A Random Pattern Extension to the Stochastic Network Pore Model, *Chem. Eng. Sci.*, **41**, 2663, (1986a)

Mann, R., Sharatt, P,N. and Thomson, G., Deactivation of a Supported Zeolite Catalyst: Diffusion Reaction and Coke Deposition in Stochastic Pore Networks, *Chem. Eng. Sci.*, **41**, 711, (1986b).

McCoy, M.A., Personal Comm., (1995)

McCoy, M.A., Hearn, B.J. and Liapis, A.I., Finite Bath Adsorption of ß-Galactosidase onto Monoclonal Antibody Ligand Immobilized on Nonporous Glass Coated Beads, *Chem. Eng. Commun.*, **108**, 225 (1991)

McCoy, M.A., Liapis, A.I. and Unger, K.K., Applications of Mathematical Modelling to the Simulation of Binary Perfusion Chromatography, *J. Chromatogr.*, **644**, 1, (1993)

Mikhail, R. Sh. and Robens, E., Eds., in "Microstructure and Thermal Analysis of Solid Surfaces", John Wiley and Sons, Chichester, UK, 175, (1983)

Miller, S.F. and King, C.J., Axial Dispersion in Liquid Flow Through Packed Beds, *A.I.ChE.J.*, **12**, 767, (1966)

Mohanty, K.K., Ottino, J.M. and Davis, H.T., Reaction and Transport in Disordered Composite Media: Introduction of Percolation Concepts, *Chem. Eng. Sci.*, **37** (6), 905, (1982).

Mojaradi, R. and Sahimi, M., Diffusional Controlled Reactions in Disordered Porous Media - II. Non- Uniform Distribution of Reactants, *Chem. Eng. Sci.*, **43**, 2995, (1988).

Nakano, Y. and Evans, J.W., Monte Carlo Simulation of Diffusion of Gases in a Porous Solid: Calculation for a New Class of Solid, *J. Chem. Phys.* **78**, 2568, (1980).

Nicholson, D. and Petropoulos J.H., Capillary Models for Porous Media: V. Flow Properties of Random Networks with Various Radius Distributions, *J. Phys. D.*, **8**, 1430, (1975)

Nicholson, D. and Petropoulos, J.H., Capillary Models for Porous Media: III. Two-Phase Flows in a Three-Dimensional Network with Gaussian Radius Distribution, *J. Phys. D.*, **4**, 181, (1971)

Niclov, R., Werner, W. and Halasz, I., Pore Size Distribution of "in-situ" Coated Silica Gels Determined by Exclusion Chromatography, *J. Chromatogr . Sci.*, **18**, 207, (1980)

Nir, A. and Pismen, L.M., Simultaneous Intraparticle Forced Convection, Diffusion, and Reaction in a Porous Catalyst, *Chem. Eng. Sci.*, **32**, 35, (1977)

Ohlson, S., Hansson, L., Larsson, P.O. and Mosbach, K., High Perfomance Liquid Affinity Chromatography (HPLAC) and its Application to the Separation of Enzymes and Antigens, *FEBS Letters,* **93**, 5, (1978)

Otani, S. and Smith, J.M., Effectiveness of Large Catalyst Pellets - An Experimental Study, *J. Catal.*, **5**, 332, (1966).

Pandey, R.B., Stauffer, D., Margolina, A. and Zabolitzky, J.G., Diffusion on Random Systems above, below and at their Percolation Threshold in Two and Three Dimensions, *J. Statistical Phys.*, **34**, 427, (1984)

Park, S. and Stephanopoulos, G.N., Packed Bed Bioreactor with Porous Ceramic Beads for Animal Cell Culture, *Biotech. Bioeng.*, **41**, 25, (1993)

Petropoulos, J.H., Petrou, J.K. and Liapis, A.I., Network Model Investigation of Gas Transport in Bi-disperse Porous Adsorbents, *Ind. Eng. Chem. Res.*, **30**, 1281, (1991)

Petrou, J.K., Petropoulos, J.H., Kanellopoulos, N.K.and Liapis, A.I., Iteractive and Renormalized Effective Medium Approximation Methods for Evaluating Transport in Bi-disperse Pore Networks, *In Fundamentals Proceedings of the Third International Conference on Fundamentals of Adsorption, May 7-12, 1989*, Santhofen, Bavaria, Federal Republic of Germany; Mersmann A.B., Scholl, S.E., Eds.; Engineering Foundation; NY, (1990)

Porath, J. and Flodin, P., Gel Filtration, A Method for Desalting and Group Separation, *Nature*, **183**, 1657, (1959)

Porath, J. and Lindner, E., Separation Methods Based on Molecular Sieving and Ion Exclusion, *Nature*, **191**, 69, (1991)

Prince, C.L., Bringi, V.and Shuler, M.L., Convective Mass Transfer in Large Porous Biocatalysts: Plant Organ Cultures, *Biotechnol. Prog.*, **7**, 195, (1991)

Probstein, R.F., in "Physicochemical Hydrodynamics - An Introduction"', Butterworths Publisher, Boston, (1989)

Rege, S.D. and Fogler, H.S., A Network Model for Deep Bed Filtration of Solid Particles and Emulsion Drops, *A.I.ChE. J.*, **34** (11) , 1761, (1988).

Rege, S.D. and Fogler, H.S., Network Model for Straining Dominated Particle Entrapment in Porous Media, *Chem. Eng. Sci.*, **42** (7), 1553, (1987).

Regnier, F. and Noel, R., Glycerolpropylsilane Bonded Phases in the Steric Exclusion Chromatography of Biological Macromolecules, *J. Chromatogr. Sci.*, **14**, 316, (1976)

Reverberi, A., Ferraiolo, G. and Peloso, A., Determinazione Sperimentale della Funzione di Distribuzone dei Macropori Cilindrici e "Ink Bottle" in Sistemi Porosi, *Ann. Chim.*, **56**, 1552, (1966)

Reyes, S. and Jensen, K.F., Percolation Concepts in Modelling of Gas-Solid Reactions - I. Applications to Char Gasification in the Kinetic Regime, *Chem. Eng. Sci.*, **41**, 333, (1986a)

Reyes, S. and Jensen, K.F., Percolation Concepts in Modelling of Gas-Solid Reactions - II. Applications to Char Gasification in the Diffusion Regime, *Chem. Eng. Sci.*, **41**, 345, (1986b)

Reyes, S. and Jensen, K.F., Percolation Concepts in Modelling of Gas-Solid Reactions - III. Applications to Sulphation of Calcined Limestone, *Chem. Eng. Sci.*, **42**, 565, (1987)

Reyes, S. and Jensen, K.J., Estimation of Effective Transport Coefficients in Porous Solids Based on Percolation Concepts, *Chem. Eng. Sci.*, **40** (9), 1723, (1985).

Reynolds, P.J., Sanley, H.E. and Klein W., Large-Scale Monte Carlo Renormalization Group for Percolation, *Phys. Rev. B*, **21**, 1223, (1980)

Ritter, H.L., and Drake, L.C., Pore-Size Distribution in Porous Materials (I) Pressure Porosimeter and Determination of Complete Macropore Size Distribution, *Ind. Eng. Chem.*, **17**, 782, (1945)

Rodrigues, A.E. and Ferreira, R.M.Q., Convection, Diffusion and Reaction in a Large-Pore Catalyst Particle, *A.I.ChE. Symp. Ser.*, **84**, 80, (1988)

Rodrigues, A.E., Ahn, B.J. and Zoulalian, A., Intraparticle-Forced Convection Effect in Catalyst Diffusivity Measurements and Reactor Design, *A.I.ChE. J.*, **28**, 541, (1982)

Rodrigues, A.E., Lopes, J.C.B., Dias, M.M. and Carta, G., Diffusion and Convection in Permeable Particles: The Analogy between Slab and Sphere Geometries, *Sep. Tech.*, **2**, 208, (1992a)

Rodrigues, A.E., Lopes, J.C.B., Lu, Z.P., Loureiro, J.M. and Dias, M.M., Importance of Intraparticle Convection in the Performance of Chromatography Processes, *J. Chromatogr.*, **590**, 93, (1992b)

Rodrigues, A.E., Lu Zuling, and Laureiro, J.M. Residence Time Distribution of Inert and Linearly Adsorbed Species in a Fixed Bed Containing Large Pore Support: Applications .Separation Engineering, *Chem. Eng, Sci.*, **46**, 2765, (1991).

Rodrigues, A.E., Lu, Z.P., Loureiro, J.M. and Carta, G., Peak Resolution in Linear Chromatography Effects of Intraparticle Convection, *J. Chromatogr.*, **653**, 189, (1993)

Rodrigues, A.E., Ramos, A.M.D., Loureiro, J.M., Diaz, M. and Lu, Z.P., Influence of Adsorption-Desorption Kinetics on the Performance of Chromatographic Processes using Large-Pore Supports, *Chem. Eng. Sci.*, **47**, 4405, (1992c)

Rose, W., Studies of Waterflood Performance - III. Use of Network Models, Illinois State Geological Survey Circular, **237,** 1, (1957).

Rumpf, H. and Gupte, A.R., Einflusse der Porositat und Korngroßenverteilung im Widerstandsgesetz der Porenstromung, *Chem. Ind. Tech.*, **43**, 367, (1971)

Ryan, D., Carbonell. R.G. and Whitaker, S., Effective Diffusivities for Caltalyst Pellets under Reactive Conditions, *Chem. Eng. Sci.*, **35**, 10, (1980).

Sahimi, M. and Imdakm, A.O., The Effect of Morphological Disorder on Hydrodynamic Dispersion in Flow Through Porous Media, *J. Phys. A.*, **21**, 3833, (1988)

Sahimi, M. and Jue, V.L., Diffusion of Large Molecules in Porous Media, *Phys. Rev. Letts,* **62** (6). 629, (1989).

Sahimi, M. and Tsotsis, T.T., Statistical Modelling of Gas-Solid Reaction with Pore Volume Growth: Kinetic Regime, *Chem. Eng. Sci.*, **43**, 113, (1988)

Sahimi, M., Davis, H.T. and Scriven, L.E., Dispersion in Disordered Media, *Chem. Eng. Comm.*, **23**, 329, (1983).

Sahimi, M., Govalas, G.R. and Tsotsis, T.T., Statistical and Continuum Models of Fluid-Solid Reactions in Porous Media, *Chem. Eng. Sci.*, **45** (6), 1443, (1990).

Sahimi, M., Heiba, A.A., Davis, H.T. and Sciven, L.E., Dispersion in Flow through Porous Media - II. Two-phase Flow, *Chem. Eng. Sci.*, **41**, 2123, (1986).

Sahimi, M., Hughes, B.D., Scriven, L.E. and Davis, H.T., Dispersion in Flow Through Porous Media. I. One-Phase Flow, *Chem. Eng. Sci.*, **41**, 2103, (1986)

Sahimi, M., in "The Mathematics and Physics of Disordered Media", B.D. Hughes and B.W. Ninham, Eds., *Lect. Notes Math.*, **1035**, 314, (1983)

Sandmann, C.W. and Zygourakis, K., Evolution of Pore Structure during Gas-Solid Reactions: Discrete Models, *Chem. Eng. Sci.*, **41**, 733, (1986)

Schechter, I., Separation of Proteins by High-Speed Pressure Liquid Chromatography, *Anal. Biochem.*, **58**, 30, (1974)

Schultz, G.V. and Baumann, H., Thermodynamisches Verhalten, Expansion skoeffizient und Viskostatszahl vol Polystyrol in Tetrahydrofuran, *Makromol. Chem.*, **114**, 122, (1968)

Scopes , R.K., "Protein Purification - Principles and Practice", Springer-Verlag, NY., 2nd Ed., (1988).

Sevick, E.M., Monson, P.A. and Ottino, J.M., Monte Carlo Calculations of Cluster Statistics in Continuum Models of Composite Morphology, *J. Chem.* Phys., **88**, 1198, (1988)

Shah, N. and Ottino J.M., Transport and Reaction in Evolving, Disordered Composites - I. Gasification of Porous Solids, *Chem. Eng. Sci.*, **42**, 63, (1987)

Sharma, M.M. and Yartsos, Y.C., Transport of Particulate Suspensions in Porous Media: Model Formulation, *A.I.ChE. J.*, **33** (10), 1636, (1987).

Sherrington, D.C., Synthesis and Characterization of Porous Polymers, *Makromol. Chem., Macromol. Symp.*, **70/71**, 303, (1993)

Shull, C.G., The Determination of Pore Size Distribution from Gas Adsorption Data, *J. Am. Chem. Soc.*, **70**, 1405, (1948)

Simons, G.A., and Finson, M.L., The Structure of Coal Char: Part I. Pore Branching, *Combust. Sci. Technol.*, **19**, 217, (1979).

Sober, H. and Peterson, E., Chromatography of Proteins on Cellulose Ion-Exchanger, *J. Am. Chem. Soc.*, **76**, 1711, (1954)

Spendley, W., Hext, G.R. and Himsworth, F.R., *Technometrics*, **4**, 44, (1962)

Stanley, H.E., in "Introduction to Phase Transitions and Critical Phenomena", Oxford University Press, NY, (1971)

Stauffer, D. and Aharony, A., in "Introduction to Percolation Theory", 2nd Edition, Taylor and Francis, Washington DC, (1992)

Stauffer, D. and Zabolitzky, J.G., Reexamination of Three Dimension Percolation Thresholds Estimates, *J. Phys. A*, **19**, 3705, (1986)

Stephanopoulos, G.N. and Tsiveriotis, K., The Effect of Intraparticle Convection on Nutrient Transport in Porous Biological Pellets, *Chem. Eng. Sci.*, **44**, 2031, (1989)

Stockmayer, W.H., Theory of Molecular Size Distribution and Gel Formation in Branched Chain Polymers, *J. Chem. Phys.*, **11**, 45, (1943).

Synder, L.R., and Kirkland, J.J., in "Introduction to Modern Liquid Chromatography", Wiley, NY., 2nd ed., (1979).

Szekely, J. and Propster, M., A Structural Model for Gas-Solid Reactions with a Moving Boundary - VI. The Effect of Grain Size Distribution on the Conversion of Porous Solids, *Chem. Eng. Sci.*, **30**, 1049, (1975).

Tanaka, N., Hashizume, K., Araki, M., Tsuchiya, H., Okuno, A., Iwaguchi, K., Ohnishi, S. and Takai, N., Internal Structures of Wide-Pore Packing Materials for High-Performance Liquid Chromatography studied by Transmission Electron Microscopy, *J. Chromatogr.*, **448**, 95, (1988)

Tanaka, N., Kimata, K., Araki, T., Tsuchiya, H. and Hashizume, K., Microscopic Characterization of High-Performance Liquid Chromatography Packing Materials, *J. Chromatogr.*, **544**, 319, (1991a)

Tanaka, N., Kimata, K., Hosoya, K., Araki, T., Tsuchiya, H. and Hashizume, K., TEM Study on the Pore Structures of Wide-Pore Silica and Polymer Gels, *J. High Resol. Chromatogr.*, **14**, 40, (1991b)

Taylor, J.K., in " Statistical Techniques for Data Analysis", Lewis Publishers, Inc., NY, (1990)

Tracz, E. and Barna, A., Preparation of Support Grains for Transmission Electron Microscopic Studies, *J. Chromatogr.*, **355**, 421, (1986)

Tracz, E., Skubiszewska, J. and Leboda, R., Microscopic Investigation of the Surface Structure of Carbon-Silica Adsorbents. I. Influence of the Modifier used on the Topography of Carbon-Silica Adsorbents, *J. Chromatogr.*, **287**, 136, (1984)

Trasz, E., Leboda, R. and Mizera, E., Microscopic Investigations of the Surface Structure of Carbon-Silica Adsorbents. III. Heterogeneity of the Carbon Layer, *J. Chromatogr.*, **355**, 412, (1986)

Tsuji, K. and Robertson, J.H., Improved HPLC Method for Polypeptide Antibiotics and its Application to Study the Effects of Treatments to Reduce Microbial Levels in Bacitracin Powder, *J. Chromatogr.*, **112**, 663, (1975)

Unger, K., Schick-Kalb, J. and Krebs, K-F., Preparation of Porous Silica Spheres for Column Liquid Chromatography, *J. Chromatogr.*, **83**, 5, (1973)

Unger, K.K. and Gimpel, M.G., Critical Assessment of Particle Size Analysis of Porous Silica Microbed HPLC Graphic Packings by Photosedimentation, *J. Chromatogr.*, **180**, 93, (1979)

Unger, K.K., Eds., in "Packings and Stationary Phases in Chromatography Techniques", Dekker, NY, (1990)

Unger, K.K., Kern, R., Ninou, M.C. and Krebs, K-F., High Performance Gel Permeation Chromatography with a New Type of Silica Packing Material, *J. Chromatogr.*, **99**, 435, (1974).

Unger, K.K., Porous Silica, *J. Chromatogr. Lib.*, **16** Elsevier, Armsterdam, 32, (1979)

Venkatasubramaniam, K. and Harrow, H.S., Design and Operation of a Commercial Immobilized Glucose Isomerase Reactor System, *Annals N.Y. Acad. Sci.*, **326**, 141, (1979).

Vilenchik, L.Z., Asrar, J., Ayotte, R.C., Ternorutsky, L. and Hardiman, C.J., Macromolecular Porosimetry, *J. Chromatogr.*, **648**, 9, (1993)

Voigt, E.M. and Tomlinson, R.H., The Determination of Pore Size Distribution and Surface Area from Adsorption Isotherms, *Can. J. Chem.*, **33**, 215, (1955)

Volkenstein, U.V., in "Configurational Statistics of Polymeric Chains", Interscience, NY., (1963)

Wakao, N. and Smith, J.M., Diffusion in Catalyst Pellets, *Chem. Eng. Sci.*, **17**, 825, (1962).

Wall, G.C. and Brown, R.J.C., The Determination of Pore-Size-Distribution from Sorption Isotherms and Mercury Penetration in Interconnected Pores: The Application of Percolation Theory, *J. Colloid Interface Sci.*, **82**, 141, (1981)

Wang, C.T. and Smith, J.M., Tortuosity Factors for Diffusion in Catalyst Pellets, *A.I.ChE. J.*, **29**, 132, (1983).

Wardlaw, N.C. and McKellar, M., Mercury Porosimetry and the Interpretation of Pore Geometry in Sedimentary Rocks and Artificial Models, *Powder Tech.*, **29**, 127, (1981)

Wardlaw, N.C., Li, Y. and Forbes, D., Pore-Throat Size Correlation from Capillary Pressure Curves, *In Transport in Porous Media*, **2**, 597, (1987)

Werner, W. and Halasz, I., Pore Structure of Chemically Modified Silica Gels Determined by Exclusion Chromatography, *J. Chromatogr. Sci.*, **18**, 277, (1980)

Wheeler, A., Presentations at Catalysis Symposium, *Am. Assoc. Adv. Sci. Conf.*, Gibson Island, June 1945, (1946)

Wilhelm, R.H., Rate Process - Application to Reactor Design, *Chem. Eng. Prog.*, **49**, 150, (1953).

Winterfield, P.H., Scriven, L.E. and Davis, H.T., Percolation and Conductivity of Random Two-Dimensional Composites, *J. Phys. C*, **14**, 2361, (1981)

Yadav, G.D., Dullien, F.A.L., Chatzis, I. and MacDonald, I.F., Microscopic Distribution of Wetting and Non-Wetting Phases in Sandstones during Immiscible Displacements, SPE 13212, (1984)

Yang, C.N., Introductory Note on Phase Transitions and Critical Phenomena, in Phase Transitions and Critical Phenomena, Damb and Lebowitz, Eds., **1**, 1, (1983)

Yau, W.W., Ginnard, C.R. and Kirkland, J.J., *J. Chromatogr.*, Broad-Range Linear Calibration in HPLC Size Exclusion Chromatography using Column Packings with Bimodal Pores, **149**, 465, (1978)

Yau, W.W., Kirkland, J.J. and Bly, D.D., in "Modern Size-Exclusion Chromatography", John Wiley and Sons, NY, (1979)

Yortsos, Y.C. and Sharma, M.M., Application of Percolation Theory to Non-Catalytic Gas-Solid Reactions, *A.I.ChE. J.*, **32**, 46, (1986)

Yu, H.C. and Sotirchos, S.V., A Generalized Pore Model for Gas-Solid Reactions Exhibiting Pore Closure, *A.I.ChE. J.*, **33**, 382, (1987)

Zallen, R., in " The Physics of Amorphous Solids", John Wiley and Sons, NY, (1983).

Zygourakis, K. and Sandmann, C.W., Discrete Structural Models and their Applications to Gas-Solid Reacting Systems, *A.I.ChE. J.*, **34**, 2030, (1988)

Appendix A

Computer Program Listings

The computer programs which were written to generate the model and to perform the simulations in this thesis are found in this appendix. The programs were written in Fortran 77 and run on the CRAY Y-MP Supercomputer.

Appendix A.1 Lattice Generation

```
      program lattice
c
c     This program sets up a representation of the POROS packed column
c
c     Declaration of variables
c
      integer siteocc(15002),siteno,m(15002),k1
      integer cluster(15002),p,q,sitebck
      integer x,y,z,orien,local,site,site1,site2,site3,site4,l
      integer siteko,sitep1,sitep2,sitep3,nperc,ninter,inter,dumint
      real mu,volcell
      real diam,bet,pi,dens,pie,ppsity,sumd,sumdcu,diamcu
      parameter (l=25,pie=3.141592654,local=3)
      integer sitelef,sitefrt,surf,localpi(17576)
      integer fill(0:l,0:l,0:l,3)
      real dia(0:l,0:l,0:l,3)
      data ninter,inter,dumint,sitep1,sitep2,sitep3/6*0/
      data site1,site2,site3,site4/4*0/
      data fill/52728*0/dia/52728*0.0/
c
c     initialize variables
c
      data n,sumdcu,sumd,global,globalpi/0,4*0.0/
      do 20 i=1,(l+1)**3
            localpi(i) = 0
20    continue
c
      nperc=nint(0.4*real(3*l*l*l))
      iseed=120560
      call rnset (iseed)
c
c     open table for random deviates of pore diameters
c
      open (5, file="pvd.rnd", status="old")
      open (6, file="pvd.inter", status="old")
      open (8, file="perc.sites",status="new")
c
c     Place pores in the spine of the lattice
c
      read(6,*) ninter
      print*,"The number of interstitial pores is: ", ninter
c
      x=12
      y=12
      do 5 z=l,1,-1
            site1=13+(l+1)*12+((l+1)**2)*z
            sitep3=13+(l+1)*12+((l+1)**2)*(z-1)
            dia(12,12,z,3)=diam
            localpi(site1)=localpi(site1)+1
            localpi(sitep3)=localpi(sitep3)+1
            inter = inter + 1
5     continue
c
c     Place interstitial pores in percolating tansect
c
      z=nint(munf()*(l-2))+1
c
3000  site1=x+1+(l+1)*y+((l+1)**2)*z
      if (localpi(site1).ge.local) call spine(l,site1,x,y,z,localpi,
      1     local,ninter,inter,no)
c
```

```
      if (dumint .eq. inter) then
            if (orien .eq. 1) then
                  orien = 2
            elseif (orien .eq. 2) then
                  orien = 3
            else
                  orien = 1
            endif
      else
            if(z.eq.(ninter-inter).and.dia(x,y,z,3).eq.0.0) then
                  orien = 3
            elseif (z.eq.(ninter-inter).and.dia(x,y,z,3).ne.0.0) then
                  call spine(l,site1,x,y,z,localpi,local,ninter,
     1                  inter,no)
                  orien=nint(munf()*2)+1
11                  if (orien.eq.3) then
                        orien=nint(munf()*2)+1
                        goto 11
                  endif
            elseif (z.ne.(ninter-inter)) then
                  orien=nint(munf()*2)+1
17                  if(x.eq.12 .and. y.eq.12 .and. orien.eq.3)then
                        orien=nint(munf()*2)+1
                        goto 17
                  endif
            endif
      endif
      sitep1=x+(l+1)*(y-1)+((l+1)**2)*z+1
      sitep2=x+2+(l+1)*y+((l+1)**2)*z
      sitep3=x+(l+1)*y+((l+1)**2)*(z-1)+1
      dumint = inter
      surf=0
c
c     check if coordinates are surface points
c
      if(x.eq.l) call interfb(surf,l,site1,site2,site3,site4,sitep1,
     2            sitep2,sitep3,localpi,local,x,y,z,orien,dia,
     3            diam,inter,ninter)
c
      if (surf .eq. 1) goto 1500
c
      if(y.eq.0) call interlr(surf,l,site1,site2,site3,site4,sitep1,
     2            sitep2,sitep3,localpi,local,x,y,z,orien,dia,
     3            diam,inter,ninter)
c
      if (surf .eq. 1) goto 1500
c
      if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local.and.
     1     localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
     2     then
            read(6,*) diam
            dia(x,y,z,orien) = diam
            inter = inter + 1
            localpi(site1)=localpi(site1)+1
            if (orien .eq. 1) then
                  localpi(sitep1)=localpi(sitep1)+1
            elseif (orien .eq. 2) then
                  localpi(sitep2)=localpi(sitep2)+1
            else
                  localpi(sitep3)=localpi(sitep3)+1
            endif
            call spine(l,site1,x,y,z,localpi,local,ninter,inter,no)
      elseif (orien.eq.1.and.localpi(sitep1).lt.local.and.dia(x,y,z,
     1     1).eq.0.0) then
```

```
                read(6,*) diam
                dia(x,y,z,orien) = diam
                inter = inter + 1
                if (y.eq.1) then
                        site2=x+1+l*(l+1)+((l+1)**2)*z
                        localpi(site2)=localpi(site2)+1
                endif
                localpi(site1)=localpi(site1)+1
                localpi(sitep1)=localpi(sitep1)+1
                y=y-1
        elseif (orien.eq.2.and.localpi(sitep2).lt.local.and.dia(x,y,z,
        1       2).eq.0.0) then
                read(6,*) diam
                dia(x,y,z,orien) = diam
                inter = inter + 1
                if (x.eq.l-1) then
                        site2=1+y*(l+1)+((l+1)**2)*z
                        localpi(site2)=localpi(site2)+1
                endif
                localpi(site1)=localpi(site1)+1
                localpi(sitep2)=localpi(sitep2)+1
                x=x+1
        elseif (orien.eq.3.and.localpi(sitep3).lt.local.and.dia(x,y,z,
        1       3).eq.0.0) then
                read(6,*) diam
                dia(x,y,z,orien) = diam
                inter = inter + 1
                localpi(site1)=localpi(site1)+1
                localpi(sitep3)=localpi(sitep3)+1
                z=z-1
                if (z.eq.0) call spine(i,site1,x,y,z,localpi,local,
        1                                       ninter,inter,no)
        endif
c
1500    continue
c
c       Check if all the interstitial pores have been added
c
        if (inter .lt. ninter) goto 3000
c
c       Add intraparticle pores randomly
c
c       Select a coordinate and choose a pore orientation
c
2000            x=nint(munf()*l)
                y=nint(munf()*l)
                z=nint(munf()*l)
15              if (z.eq.0) then
                        z=nint(munf()*l)
                        goto 15
                endif
                orien=nint(munf()*2)+1
                site1 = x+(l+1)*y+((l+1)**2)*z+1
                sitep1= x+(l+1)*(y-1)+((l+1)**2)*z+1
                sitep2= x+2+(l+1)*y+((l+1)**2)*z
                sitep3= x+(l+1)*y+((l+1)**2)*(z-1)+1
                dumloca = localpi(site1)
c
c       check if coordinates selected are surface points
c
        if ((x.eq.0 .or. x.eq.l) .and. (localpi(site1).lt.local .and.
        1       dia(x,y,z,orien).eq.0.0)) call fbsurf(l,site1,site2,
        2       site3,site4,sitep1,sitep2,sitep3,localpi,local,x,y,z,
        3       orien,dia,diam,diamcu,mu)
```

```
c
      if (localpi(site1) .ne. dumloca) goto 1000
c
c
      if ((y.eq.0 .or. y.eq.l) .and. (localpi(site1).lt.local .and.
     1      dia(x,y,z,orien).eq.0.0)) call lrsurf(l,site1,site2,
     2      site3,site4,sitep1,sitep2,sitep3,localpi,local,x,y,z,
     3      orien,dia,diam,diamcu,mu)
c
      if (localpi(site1) .ne. dumloca) goto 1000
c
      if (z.eq.l) orien=3
      if((localpi(site1).lt.local).and.(dia(x,y,z,orien).eq.0.0))then
            if((orien.eq.1).and.(localpi(sitep1).lt.local)) then
                  read(5,*) diam
                  dia(x,y,z,orien) = diam
                  if (y.eq.1) then
                        site2=x+1+l*(l+1)+((l+1)**2)*z
                        localpi(site2)=localpi(site2)+1
                  endif
                  localpi(site1) = localpi(site1)+1
                  localpi(sitep1)=localpi(sitep1)+1
            elseif((orien.eq.2).and.(localpi(sitep2).lt.local)) then
                  read(5,*) diam
                  dia(x,y,z,orien) = diam
                  if (x.eq.l-1) then
                        site2=1+y*(l+1)+((l+1)**2)*z
                        localpi(site2)=localpi(site2)+1
                  endif
                  localpi(site1) = localpi(site1)+1
                  localpi(sitep2)=localpi(sitep2)+1
            elseif((orien.eq.3).and.(localpi(sitep3).lt.local)) then
                  read(5,*) diam
                  dia(x,y,z,orien) = diam
                  localpi(site1) = localpi(site1)+1
                  localpi(sitep3)=localpi(sitep3)+1
            endif
      endif
c
1000  continue
      if (localpi(site1) .ne. dumloca) then
            n=n+1
      endif
c
c     check if enough pores have been added
c
      if(n .lt. (nperc-ninter)) goto 2000
c
      globalpi = global/real(site)
      print*, "Global PI is ",globalpi
c
      call hoshen(l,dia,localpi,siteocc)
c
c     Pick out the dead end sites, that is, sites which has a local
c     pore-interconnectivity of 1
c     These sites will eliminate isolated groups of pores from all
c     hydrodynamic calculations
c
      do 300 z=l-1,1,-1
      do 300 x=0,l-1
      do 300 y=0,l-1
            siteko=x+1+y*(l+1)+((l+1)**2)*z
            siteno=x*l+y+2+(l-1-z)*l*l
            if (localpi(siteko).eq.1) then
```

```
                        siteno=x*l+y+2+(l-1-z)*l*l
                        siteocc(siteno)=siteno
                endif
300     continue
c
        k1=0
        do 200 i=2,15002
                if (siteocc(i) .ne. 0) then
                        k1=k1+1
                        m(k1)=i
                        write(8,*)m(k1)
                endif
200     continue
c
        print*,"The number of dead sites is:  ",k1
        stop
        end

        subroutine fbsurf(l,site1,site2,site3,site4,sitep1,sitep2,sitep3,
        1       localpi,local,x,y,z,orien,dia,diam,diamcu,mu)
        integer x,y,z,orien,local,site1,site2,site3,site4,l
        integer localpi(17576),sitep1,sitep2,sitep3
        integer site5,site6,site7,site8
        real dia(0:l,0:l,0:l,3)
        real diam,diamcu,mu
c
c
        open (5, file="pvd.rnd", status="old")
c
        if (z.eq.l) orien=3
c
c       take care of the periodic boundary conditions of the four vertical
c       edges
c
        if (x.eq.0) then
                site2=l+1+y*(l+1)+((l+1)**2)*z
        else
                site2=1+y*(l+1)+((l+1)**2)*z
        endif
c
        if (orien.eq.1 .and. (y.eq.0 .or. y.eq.l)) then
                if (y.eq.0) then
                        site3=1+(l+1)*l+((l+1)**2)*z
                        site4=l+1+l*(l+1)+((l+1)**2)*z
                elseif (y.eq.l) then
                        site3=1+((l+1)**2)*z
                        site4=l+1+((l+1)**2)*z
                endif
                        site5=1+(l+1)*(l-1)+((l+1)**2)*z
                        site6=l+1+(l+1)*(l-1)+((l+1)**2)*z
                if (localpi(site5).lt.local) then
                        read (5,*) diam
                        dia(0,0,z,1)=diam
                        dia(l,0,z,1)=diam
                        dia(0,l,z,1)=diam
                        dia(l,l,z,1)=diam
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        localpi(site5)=localpi(site5)+1
                        localpi(site6)=localpi(site6)+1
                endif
c
```

```
      elseif (orien.eq.2 .and. (y.eq.0 .or. y.eq.l)) then
            if (y.eq.0) then
                  site3=1+(l+1)*l+((l+1)**2)*z
                  site4=l+1+l*(l+1)+((l+1)**2)*z
            elseif (y.eq.l) then
                  site3=((l+1)**2)*z+1
                  site4=l+1+((l+1)**2)*z
            endif
                  site5=2+((l+1)**2)*z
                  site6=2+(l+1)*l+((l+1)**2)*z
            if (localpi(site5).lt.local) then
                  read (5,*) diam
                  dia(0,0,z,2)=diam
                  dia(l,0,z,2)=diam
                  dia(0,l,z,2)=diam
                  dia(l,l,z,2)=diam
                  localpi(site1)=localpi(site1)+1
                  localpi(site2)=localpi(site2)+1
                  localpi(site3)=localpi(site3)+1
                  localpi(site4)=localpi(site4)+1
                  localpi(site5)=localpi(site5)+1
                  localpi(site6)=localpi(site6)+1
            endif
c
      elseif (orien.eq.3 .and. (y.eq.0 .or. y.eq.l)) then
            if (y.eq.0) then
                  site3=1+(l+1)*l+((l+1)**2)*z
                  site4=l+1+l*(l+1)+((l+1)**2)*z
            elseif (y.eq.l) then
                  site3=1+((l+1)**2)*z
                  site4=l+1+((l+1)**2)*z
            endif
                  site5=1+((l+1)**2)*(z-1)
                  site6=l+1+((l+1)**2)*(z-1)
                  site7=l*(l+1)+((l+1)**2)*(z-1)
                  site8=l+1+l*(l+1)+((l+1)**2)*(z-1)
            if (localpi(site5).lt.local) tnen
                  read (5,*) diam
                  dia(0,0,z,3)=diam
                  dia(l,0,z,3)=diam
                  dia(0,l,z,3)=diam
                  dia(l,l,z,3)=diam
                  localpi(site1)=localpi(site1)+1
                  localpi(site2)=localpi(site2)+1
                  localpi(site3)=localpi(site3)+1
                  localpi(site4)=localpi(site4)+1
                  localpi(site5)=localpi(site5)+1
                  localpi(site6)=localpi(site6)+1
                  localpi(site7)=localpi(site7)+1
                  localpi(site8)=localpi(site8)+1
            endif
      endif
c
      if (x.eq.0)then
            if((orien .eq. 2).and.(localpi(sitep2).lt.local))then
                        read(5,*) diam
                        dia(0,y,z,orien)=diam
                        dia(l,y,z,orien)=diam
                        site3= 2+(l+1)*y+((l+1)**2)*z
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
            elseif((orien.eq.1).and.(localpi(sitep1).lt.local))then
                        read(5,*) diam
```

```
                        dia(0,y,z,orien)=diam
                        dia(l,y,z,orien)=diam
                        site3= (l+1)*(y-1)+((l+1)**2)*z+1
                        site4= l+1+(l+1)*(y-1)+((l+1)**2)*z
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
            elseif((orien.eq.3).and.(localpi(sitep3).lt.local))then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(l,y,z,orien)=diam
                        site3= (l+1)*y+((l+1)**2)*(z-1)+1
                        site4= l+1+(l+1)*y+((l+1)**2)*(z-1)
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
            endif
c
c
      else
            if((orien.eq.1).and.(localpi(sitep1).lt.local)) then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        site3= l+1+(l+1)*(y-1)+((l+1)**2)*z
                        site4= (l+1)*(y-1)+((l+1)**2)*z+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
            elseif((orien.eq.3).and.(localpi(sitep3).lt.local))then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        site3= l+1+(l+1)*y+((l+1)**2)*(z-1)
                        site4= (l+1)*y+((l+1)**2)*(z-1)+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
            elseif (orien.eq.2) then
                        sitep2=2+(l+1)*y+((l+1)**2)*z
                  if(localpi(sitep2).lt.local) then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(sitep2)=localpi(sitep2)+1
                  endif
            endif
c
      endif
c
      return
      end


      subroutine lrsurf(l,site1,site2,site3,site4,sitep1,sitep2,sitep3,
     1      localpi,local,x,y,z,orien,dia,diam,diamcu,mu)
      integer x,y,z,orien,local,site1,site2,site3,site4,l
      integer localpi(17576),sitep1,sitep2,sitep3
```

```
      real dia(0:l,0:l,0:l,3)
      real diam,diamcu,mu
c
c
      open (5, file="pvd.rnd", status="old")
c
      if (z.eq.l) orien=3
c
      if (y.eq.0) then
            site2 = x+(l+1)*l+((l+1)**2)*z+1
c
c
            if((orien.eq.2).and.(localpi(sitep2).lt.local)) then
                        site3= x+2+((l+1)**2)*z
                        site4= x+2+(l+1)*l+((l+1)**2)*z
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
             elseif((orien.eq.3).and.(localpi(sitep3).lt.local))then
                        site3= x+1+((l+1)**2)*(z-1)
                        site4= x+1+(l+1)*l+((l+1)**2)*(z-1)
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
             elseif(orien.eq.1) then
                        sitep1=x+1+(l+1)*(l-1)+((l+1)**2)*z
                  if (localpi(sitep1).lt.local) then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(sitep1)=localpi(sitep1)+1
                  endif
             endif
c
      else
            site2 = (x+1)+((l+1)**2)*z
c
c
            if((orien.eq.1).and.(localpi(sitep1).lt.local)) then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,0,z,orien)=diam
                        site3= x+1+(l+1)*(y-1)+((l+1)**2)*z
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
             elseif((orien.eq.2).and.(localpi(sitep2).lt.local))then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,0,z,orien)=diam
                        site3= x+2+(l+1)*y+((l+1)**2)*z
                        site4= x+2+((l+1)**2)*z
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
```

```
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                elseif((orien.eq.3).and.(localpi(sitep3).lt.local))then
                        read(5,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,0,z,orien)=diam
                        site3= x+1+(l+1)*y+((l+1)**2)*(z-1)
                        site4= x+1+((l+1)**2)*(z-1)
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                endif
c
        endif
c
        return
        end



        subroutine interfb(surf,l,site1,site2,site3,site4,sitep1,sitep2,
        1       sitep3,localpi,local,x,y,z,orien,dia,diam,inter,
        2       ninter)
        integer x,y,z,orien,local,site1,site2,site3,site4,l
        integer surf,localpi(17576),sitep1,sitep2,sitep3
        integer site5,site6,site7,site8,inter,ninter
        real dia(0:l,0:l,0:l,3)
        real diam
c
c
        open (6, file="pvd.inter", status="old")
c
c       take care of the periodic boundary conditions of the four vertical
c       edges
c
        surf = 1
        if (x.eq.0) then
                site2=l+1+(l+1)*y+((l+1)**2)*z
        else
                site2=y*(l+1)+((l+1)**2)*z+1
        endif
c
        if (y.eq.0) then
                sitep1=1+(l+1)*(l-1)+((l+1)**2)*z
                sitep2=2+(l+1)*l+((l+1)**2)*z
                sitep3=1+(l+1)*l+((l+1)**2)*(z-1)
                site2=((l+1)**2)*z
        endif
c
        if (orien.eq.1 .and. (y.eq.0 .or. y.eq.l)) then
                if (y.eq.0) then
                        site3=1+(l+1)*l+((l+1)**2)*z
                        site4=l+1+l*(l+1)+((l+1)**2)*z
                elseif (y.eq.l) then
                        site3=1+((l+1)**2)*z
                        site4=l+1+((l+1)**2)*z
                endif
                site5=1+(l+1)*(l-1)+((l+1)**2)*z
                site6=l+1+(l+1)*(l-1)+((l+1)**2)*z
                if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
        1       .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
        2       then
                        read (6,*) diam
```

```
                dia(0,0,z,1)=diam
                dia(l,0,z,1)=diam
                dia(0,l,z,1)=diam
                dia(l,l,z,1)=diam
                inter = inter + 1
                localpi(site1)=localpi(site1)+1
                localpi(site2)=localpi(site2)+1
                localpi(site3)=localpi(site3)+1
                localpi(site4)=localpi(site4)+1
                localpi(site5)=localpi(site5)+1
                localpi(site6)=localpi(site6)+1
                call spine(l,site1,x,y,z,localpi,local,ninter,inter,no)
        elseif (localpi(sitep1).lt.local.and.dia(x,y,z,orien).eq.
1               0.0) then
                read (6,*) diam
                dia(0,0,z,1)=diam
                dia(l,0,z,1)=diam
                dia(0,l,z,1)=diam
                dia(l,l,z,1)=diam
                inter = inter + 1
                localpi(site1)=localpi(site1)+1
                localpi(site2)=localpi(site2)+1
                localpi(site3)=localpi(site3)+1
                localpi(site4)=localpi(site4)+1
                localpi(site5)=localpi(site5)+1
                localpi(site6)=localpi(site6)+1
                y=l-1
        endif
c
    elseif (orien.eq.2 .and. (y.eq.0 .or. y.eq.l)) then
        if (y.eq.0) then
                site3=1+(l+1)*l+((l+1)**2)*z
                site4=l+1+l*(l+1)+((l+1)**2)*z
        elseif (y.eq.l) then
                site3=((l+1)**2)*z+1
                site4=l+1+((l+1)**2)*z
        endif
        site5=2+((l+1)**2)*z
        site6=2+(l+1)*l+((l+1)**2)*z
        if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
1       .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
2       then
                read (6,*) diam
                dia(0,0,z,2)=diam
                dia(l,0,z,2)=diam
                dia(0,l,z,2)=diam
                dia(l,l,z,2)=diam
                inter = inter + 1
                localpi(site1)=localpi(site1)+1
                localpi(site2)=localpi(site2)+1
                localpi(site3)=localpi(site3)+1
                localpi(site4)=localpi(site4)+1
                localpi(site5)=localpi(site5)+1
                localpi(site6)=localpi(site6)+1
                call spine(l,site1,x,y,z,localpi,local,ninter,inter,no)
        elseif (localpi(sitep2) .lt. local.and.dia(x,y,z,orien).eq.
1                       0.0) then
                read (6,*) diam
                dia(0,0,z,2)=diam
                dia(l,0,z,2)=diam
                dia(0,l,z,2)=diam
                dia(l,l,z,2)=diam
                inter = inter + 1
                localpi(site1)=localpi(site1)+1
```

```
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        localpi(site5)=localpi(site5)+1
                        localpi(site6)=localpi(site6)+1
                        x=1
                endif
c
        elseif (orien.eq.3 .and. (y.eq.0 .or. y.eq.l)) then
                if (y.eq.0) then
                        site3=1+(l+1)*l+((l+1)**2)*z
                        site4=l+1+l*(l+1)+((l+1)**2)*z
                elseif (y.eq.l) then
                        site3=1+((l+1)**2)*z
                        site4=l+1+((l+1)**2)*z
                endif
                site5=1+((l+1)**2)*(z-1)
                site6=l+1+((l+1)**2)*(z-1)
                site7=l*(l+1)+((l+1)**2)*(z-1)
                site8=l+1+l*(l+1)+((l+1)**2)*(z-1)
                if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
        1       .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
        2       then
                        read (6,*) diam
                        dia(0,0,z,3)=diam
                        dia(l,0,z,3)=diam
                        dia(0,l,z,3)=diam
                        dia(l,l,z,3)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        localpi(site5)=localpi(site5)+1
                        localpi(site6)=localpi(site6)+1
                        localpi(site7)=localpi(site7)+1
                        localpi(site8)=localpi(site8)+1
                        call spine(l,site1,x,y,z,localpi,local,ninter,inter,no)
                elseif (localpi(sitep3) .lt. local.and.dia(x,y,z,orien).eq.
        1               0.0) then
                        read (6,*) diam
                        dia(0,0,z,3)=diam
                        dia(l,0,z,3)=diam
                        dia(0,l,z,3)=diam
                        dia(l,l,z,3)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        localpi(site5)=localpi(site5)+1
                        localpi(site6)=localpi(site6)+1
                        localpi(site7)=localpi(site7)+1
                        localpi(site8)=localpi(site8)+1
                        z=z-1
                        if (z.eq.0) call spine(l,site1,x,y,z,localpi,local,
        1                                       ninter,inter,no)
                endif
        endif
c
c
        if (y.ne.0 .or. y.ne.l) then
                site2 = (l+1)*y+((l+1)**2)*z+1
c
```

```
      if(orien.eq.1) then
      if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
     1  .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
     2  then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        inter = inter + 1
                        site3= l+1+(l+1)*(y-1)+((l+1)**2)*z
                        site4= (l+1)*(y-1)+((l+1)**2)*z+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        call spine(l,site1,x,y,z,localpi,local,
     1                        ninter,inter,no)
      elseif (localpi(sitep1) .lt.local.and.dia(x,y,z,orien).eq.
     1          0.0) then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        inter = inter + 1
                        site3= l+1+(l+1)*(y-1)+((l+1)**2)*z
                        site4= (l+1)*(y-1)+((l+1)**2)*z+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        y=y-1
      endif
      elseif(orien.eq.3) then
      if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
     1  .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
     2  then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        inter = inter + 1
                        site3= l+1+(l+1)*y+((l+1)**2)*(z-1)
                        site4= (l+1)*y+((l+1)**2)*(z-1)+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        call spine(l,site1,x,y,z,localpi,local,
     1                        ninter,inter,no)
      elseif (localpi(sitep3) .lt. local.and.dia(x,y,z,orien).eq.
     1          0.0) then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(0,y,z,orien)=diam
                        inter = inter + 1
                        site3= l+1+(l+1)*y+((l+1)**2)*(z-1)
                        site4= (l+1)*y+((l+1)**2)*(z-1)+1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        z=z-1
                        if (z.eq.0)call spine(l,site1,x,y,z,localpi,
     1                        local,ninter,inter,no)
      endif
      elseif (orien.eq.2) then
                        sitep2=2+(l+1)*y+((l+1)**2)*z
```

```
            if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
     1      .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
     2      then
                              read(6,*) diam
                              dia(x,y,z,orien)=diam
                              dia(0,y,z,orien)=diam
                              inter = inter + 1
                              localpi(site1)=localpi(site1)+1
                              localpi(site2)=localpi(site2)+1
                              localpi(sitep2)=localpi(sitep2)+1
                              call spine(l,site1,x,y,z,localpi,local,
     1                                ninter,inter,no)
            elseif (localpi(sitep2) .lt. local.and.dia(x,y,z,orien).eq.
     1                        0.0) then
                              read(6,*) diam
                              dia(x,y,z,orien)=diam
                              dia(0,y,z,orien)=diam
                              inter = inter + 1
                              localpi(site1)=localpi(site1)+1
                              localpi(site2)=localpi(site2)+1
                              localpi(sitep2)=localpi(sitep2)+1
                              x=1
            endif
            endif
c
      endif
c
      return
      end


      subroutine interlr(surf,l,site1,site2,site3,site4,sitep1,sitep2,
      1       sitep3,localpi,local,x,y,z,orien,dia,diam,inter,
      2       ninter)
      integer x,y,z,orien,local,site1,site2,site3,site4,l
      integer surf,localpi(17576),sitep1,sitep2,sitep3,inter,ninter
      real dia(0:l,0:l,0:l,3)
      real diam
c
c
      open (6, file="pvd.inter", status="old")
c
      surf = 1
      sitep1=x+(l+1)*(l-1)+((l+1)**2)*z+1
      if (y.eq.0) then
            site2 = x+(l+1)*l+((l+1)**2)*z+1
c
            if(orien.eq.2) then
                              site3= x+2+((l+1)**2)*z
                              site4= x+2+(l+1)*l+((l+1)**2)*z
            if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
     1      .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
     2      then
                              read(6,*) diam
                              dia(x,y,z,orien)=diam
                              dia(x,l,z,orien)=diam
                              inter = inter + 1
                              localpi(site1)=localpi(site1)+1
                              localpi(site2)=localpi(site2)+1
                              localpi(site3)=localpi(site3)+1
                              localpi(site4)=localpi(site4)+1
                              call spine(l,site1,x,y,z,localpi,local,
     1                                ninter,inter,no)
            elseif (localpi(sitep2) .lt. local.and.dia(x,y,z,orien).eq.
```

```
1                0.0) then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        x=x+1
        endif
        elseif(orien.eq.3) then
                        site3= x+1+((l+1)**2)*(z-1)
                        site4= x+1+(l+1)*l+((l+1)**2)*(z-1)
        if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
1       .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
2       then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        call spine(l,site1,x,y,z,localpi,local,
1                               ninter,inter,no)
        elseif (localpi(sitep3) .lt. local.and.dia(x,y,z,orien).eq.
1               0.0) then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(site3)=localpi(site3)+1
                        localpi(site4)=localpi(site4)+1
                        z=z-1
                        if (z.eq.0) call spine(l,site1,x,y,z,localpi,
1                               local,ninter,inter,no)
        endif
        elseif(orien.eq.1) then
        if (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local
1       .and.localpi(sitep3).ge.local.and.dia(x,y,z,orien).eq.0.0)
2       then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(sitep1)=localpi(sitep1)+1
                        call spine(l,site1,x,y,z,localpi,local,
1                               ninter,inter,no)
        elseif (localpi(sitep1) .lt. local.and.dia(x,y,z,orien).eq.
1               0.0) then
                        read(6,*) diam
                        dia(x,y,z,orien)=diam
                        dia(x,l,z,orien)=diam
                        inter = inter + 1
                        localpi(site1)=localpi(site1)+1
                        localpi(site2)=localpi(site2)+1
                        localpi(sitep1)=localpi(sitep1)+1
                        y=l-1
```

```
                endif
                endif
        endif
c
        return
        end

        subroutine spine(l,site1,x,y,z,localpi,local,ninter,inter,no)
c
        integer x,y,z,localpi(17576),site1,local,ninter,inter
        integer sitep1,sitep2
c
        x=12
        y=12
        z=nint(rnunf()*(l-2))+1
c
        site1=13+(l+1)*12+((l+1)**2)*z
        sitep1=x+(l+1)*(y-1)+((l+1)**2)*z+1
        sitep2=x+2+(l+1)*y+((l+1)**2)*z
        if(ninter .eq. inter) return
c
13      if (localpi(site1).ge.local.or.z.ge.(ninter-inter).or.
        1       (localpi(sitep1).ge.local.and.localpi(sitep2).ge.local))
        2       then
                z=nint(rnunf()*(l-2))+1
                site1=13+(l+1)*12+((l+1)**2)*z
                goto 13
        endif
c
        no = no+1
c
        return
        end

        program cumul
c
c       This program generates the cumulative pore size distribution
c       using a sum of normal distributions in two files:
c       pfile(1) for the pore size distribution and
c       pfile(2) for the cumulative distribution while the parameters
c       for the pore distributions are stored in pfile(3)
c
        real normal, p(4), mu(4), sigma(4), asum, asumt, prop, mean
        real lnormal,stddev, norm, cum, cumpsd, dia
c
        character*6 pfile(3)
c
        data  asum,asumt,cum,cumpsd,norm,p,mu,sigma,dia/18*0.0/
c
        normal (prop,mean,stddev,dia) =prop*exp(-0.5*((dia-mean)/
        1       stddev)**2.0)/(stddev*2.50662828)
c
        print*,"Enter the PSD filename and CumPSD filename:"
        read(5,100) pfile(1), pfile(2)
        print*,"Enter the Parameter filename:"
        read(5,200) pfile(3)
c
        open(6, file=pfile(1), status="new")
        open(7, file=pfile(2), status="new")
        open(8, file=pfile(3), status="new")
c
        print*,"Enter proportion, mean and std dev. for micropores:"
        read*, p(1), mu(1), sigma(1)
        write(8,*) "Micropore Parameters:"
```

```
      write(8,*) "--------------------"
      write(8,*) "Prop ",p(1), "  Mean ",mu(1), "  Dev ",sigma(1)
      write(8,*) " "
c
      print*,"Enter proportion, mean and std dev. for macropores:"
      read*, p(2), mu(2), sigma(2)
      write(8,*) "Macropore I Parameters:"
      write(8,*) "----------------------"
      write(8,*) "Prop ",p(2), "  Mean ",mu(2), "  Dev ",sigma(2)
      write(8,*) " "
c
      print*,"Enter proportion, mean and std dev. for Inter1 pores:"
      read*, p(3), mu(3), sigma(3)
      write(8,*) "Macropore II Parameters:"
      write(8,*) "-----------------------"
      write(8,*) "Prop ",p(3), "  Mean ",mu(3), "  Dev ",sigma(3)
      write(8,*) " "
c
      print*,"Enter proportion, mean and std dev. for Inter2 pores:"
      read*, p(4), mu(4), sigma(4)
      write(8,*) "Interstitial Pore Parameters:"
      write(8,*) "-----------------------------"
      write(8,*) "Prop ",p(4), "  Mean ",mu(4), "  Dev ",sigma(4)
      write(8,*) " "
c
      write(6,*) 0.0025, 0.0
c
      do 11 dia=0.003, 2.0, 0.001
            asum=0.0
            do 10 i=1,4
                  prop=p(i)
                  mean=mu(i)
                  stddev=sigma(i)
c
                  norm=normal(prop,mean,stddev,dia)
                  asum = asum + norm
10          continue
            asumt = asumt + asum
            write (6,*) dia, asum
11    continue
c
      do 21 dia=2.2, 19.8, 0.2
            asum=0.0
            do 20 i=1,4
                  prop=p(i)
                  mean=mu(i)
                  stddev=sigma(i)
c
                  norm=normal(prop,mean,stddev,dia)
                  asum = asum + norm
20          continue
            asumt = asumt + asum
            write (6,*) dia, asum
21    continue
c
      do 31 dia=20., 40., 1.0
            asum=0.0
            do 30 i=1,4
                  prop=p(i)
                  mean=mu(i)
                  stddev=sigma(i)
c
                  norm=normal(prop,mean,stddev,dia)
                  asum = asum + norm
```

```
30              continue
                asumt = asumt + asum
                write (6,*) dia, asum
31      continue
c
c
        close(6, status="keep")
        asum = 0.0
        open(6, file=pfile(1), status="old")
c
        do 50 j=1,2109
                read (6,*) dia, asum
                cum = cum + asum
                cumpsd = cum/asumt
                write (7,*) dia, cumpsd
50      continue
c
100     format (a6/a6)
200     format (a6)
        stop
        end

        program table1
c
c       This program prepares a table of random deviates used in
c       setting up lattice
c
        integer ninter,ndata, iopt, ldtabl, iseed
        parameter (ltabl=2075, nr=19000)
        real dia, table (ltabl,5), r(nr)
        character*10 pfile(4)
c
c       Files need to be opened for I/O purposes
c
        print*,"Enter the cumulative PSD filename:"
        read(5,200) pfile(1)
c
        print*,"Enter the pore diameter filename:"
        read(5,300) pfile(2)
c
        print*,"Enter the interstitial filename:"
        read(5,300) pfile(3)
c
        print*,"Enter the pvd#.rnd filename:"
        read(5,300) pfile(4)
c
        open (6, file=pfile(1), status="old")
        open (7, file=pfile(2), status="new")
        open (8, file=pfile(3), status="new")
        open (9, file=pfile(4), status="new")
c
        iopt = 1
c
c       Pore diameter and cumulative probabilities are read for interpolation
c       and random number generation
c
        do 10 i= 1,ltabl
                read (6,*) table(i,1), table(i,2)
10      continue
c
c       A table is generated internally to prepare for the generation of
c       random deviates
c
        call rngcs (cdf, iopt, ltabl, table, ltabl)
```

```
c
c       The seed for the random number generator is warmed up
c
        iseed=120560
        call rnset (iseed)
        call rnget (iseed)
c
c       Pore diameters are randomly generated and entered into a new table
c
        call rngct (nr, ltabl, table, ltabl, r)
        write (7,100) r
        close (7, status="keep")
c
        open (7, file=pfile(2), status="old")
        do 20 i=1,nr
                read(7,100) dia
20              if (dia.ge.1.0) ninter = ninter + 1
c
        close (7, status="keep")
        open (7, file=pfile(2), status="old")
        write (8,*) ninter
c
        do 30 i=1,nr
                read(7,100) dia
                if (dia.ge.1.0) then
                        write(8,100) dia
                else
                        write(9,100) dia
                endif
30      continue
c
100     format (/,f9.5)
200     format (a6)
300     format (a10)
c
        stop
        end

        subroutine hoshen(l,dia,localpi,siteocc)
c
c       This program locates the percolating cluster (1) using the
c       Hoshen-Kopelman algorithm of multiple cluster labelling
c
        real dia(0:l,0:l,0:l,3)
        integer siteko,siteno,cluster(15002),maxi,up,left,back
        integer orien,p,q,lptr(15002),la,ms,nofs,right,front
        integer mini,siteup,sitelef,siterit,sitefrt,sitebck
        integer c1,x,y,z,localpi(17576),siteocc(15002)
c
        c1=1
        maxi=15002
        do 5 i=1,(l-1)*l*l+2
                if(i.eq.1 .or. i.eq.(l-1)*l*l+2) then
                        cluster(i)=1
                else
                        cluster(i)=maxi
                endif
                lptr(i)=1
5       continue
c
        do 10 z=l-1,1,-1
        do 10 x=0,l-1
        do 10 y=0,l-1
                siteko=x+(l+1)*y+((l+1)**2)*z+1
```

```
                if (localpi(siteko).eq.0) goto 10
                if(x.eq.0 .and. y.eq.l-1) goto 99
                if(x.eq.l-1 .and. y.eq.0) goto 101
                if(x.eq.l-1 .and. y.eq.l-1) goto 103
        if(x.eq.0 .and. y.eq.0) then
                if(dia(0,0,z+1,3).ne.0.0) then
c
c       Search only the top connection
c
                        if(z.eq.l-1) then
                                siteup=1
                        else
                                siteup=2+(l-z-2)*l*l
                        endif
                        up=cluster(siteup)
c
                        if(lptr(up).gt.0) goto 11
                        ms=lptr(up)
12                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 12
                        lptr(up)=-la
                        up=la
c
11                      mini=up
c
                        nofs=1
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                        siteno=2+(l-1-z)*l*l
                        cluster(siteno)=mini
                        lptr(mini)=nofs
                else
                        c1=c1+1
                        siteno=2+(l-1-z)*l*l
                        cluster(siteno)=c1
                        lptr(c1)=1
                endif
                goto 10
        endif
c
        if(x.eq.0) then
                if(dia(0,y,z,1).ne.0.0 .or. dia(0,y,z+1,3).ne.0.0) then
c
c       Search the top connection
c
                        if(dia(0,y,z+1,3).eq.0.0) then
                                up=maxi
                                goto 13
                        else
c
                                if(z.eq.l-1) then
                                        siteup=1
                                else
                                        siteup=y+2+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
c
                        if(lptr(up).gt.0) goto 13
                        ms=lptr(up)
14                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 14
```

```
                        lptr(up)=-la
                        up=la
c
c       Search the left connection
c
13                      if(dia(0,y,z,1).eq.0.0) then
                                left=maxi
                                goto 15
                        else
                                sitelef=y+1+(l-1-z)*l*l
                                left=cluster(sitelef)
                        endif
c
                        if(lptr(left).gt.0) goto 15
                        ms=lptr(left)
16                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 16
                        lptr(left)=-la
                        left=la
c
15                      mini=min0(up,left)
c
                        nofs=1
                        if(up.eq.left) goto 17
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
c
17                      if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
c
                        siteno=y+2+(l-1-z)*l*l
                        cluster(siteno)=mini
                        lptr(mini)=nofs
                else
                        c1=c1+1
                        siteno=y+2+(l-1-z)*l*l
                        cluster(siteno)=c1
                        lptr(c1)=1
                endif
                goto 10
        endif
c
99      if(x.eq.0 .and. y.eq.l-1) then
                if(dia(0,l-1,z,1).ne.0.0 .or. dia(0,l-1,z+1,3).ne.0.0
        1       .or. dia(0,0,z,1).ne.0.0) then
c
c       Search the left connection
c
                        if(dia(0,l-1,z,1).eq.0.0) then
                                left=maxi
                                goto 18
                        else
                                sitelef=l+(l-1-z)*l*l
                                left=cluster(sitelef)
                        endif
c
                        if(lptr(left).gt.0) goto 18
                        ms=lptr(left)
19                      la=-ms
```

```
                        ms=lptr(la)
                        if(ms.lt.0) goto 19
                        lptr(left)=-la
                        left=la
c
c       Search the top connection
c
18                      if(dia(0,l-1,z+1,3).eq.0.0) then
                                up=maxi
                                goto 20
                        else
                                if(z.eq.l-1) then
                                        siteup=1
                                else
                                        siteup=l+1+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
c
                        if(lptr(up).gt.0) goto 20
                        ms=lptr(up)
21                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 21
                        lptr(up)=-la
                        up=la
c
c       Search the right connection
c
20                      if(dia(0,0,z,1).eq.0.0) then
                                right=maxi
                                goto 22
                        else
                                siterit=2+(l-1-z)*l*l
                                right=cluster(siterit)
                        endif
c
                        if(lptr(right).gt.0) goto 22
                        ms=lptr(right)
23                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 23
                        lptr(right)=-la
                        right=la
c
22                      mini=min0(left,up,right)
c
                        nofs=1
                        if(left.eq.up .and. up.eq.right) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                        elseif(up.eq.left) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(right.ne.maxi) then
                                        nofs=nofs+lptr(right)
                                        lptr(right)=-mini
                                endif
                        elseif(up.eq.right) then
                                if(up.ne.maxi) then
```

```
                                        ncfs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                        elseif(left.eq.right) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                        else
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                                if(right.ne.maxi) then
                                        nofs=nofs+lptr(right)
                                        lptr(right)=-mini
                                endif
                        endif
                        siteno=l+1+(l-1-z)*l*l
                        cluster(siteno)=mini
                        lptr(mini)=nofs
                else
                        c1=c1+1
                        siteno=l+1+(l-1-z)*l*l
                        cluster(siteno)=c1
                        lptr(c1)=1
                endif
                goto 10
        endif
c
        if(y.eq.0) then
                if(dia(x,0,z+1,3).ne.0.0 .or. dia(x-1,0,z,2).ne.0.0) then
c
c       Search the top connection
c
                        if(dia(x,0,z+1,3).eq.0.0) then
                                up=maxi
                                goto 25
                        else
                                if(z.eq.l-1) then
                                        siteup=1
                                else
                                        siteup=x*l+2+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
c
                        if(lptr(up).gt.0) goto 25
                        ms=lptr(up)
26                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 26
```

```
                        lptr(up)=-la
                        up=la
c
c       Search the back connection
c
25                      if(dia(x-1,0,z,2).eq.0.0) then
                                back=maxi
                                goto 27
                        else
                                sitebck=(x-1)*l+2+(l-1-z)*l*l
                                back=cluster(sitebck)
                        endif
c
                        if(lptr(back).gt.0) goto 27
                        ms=lptr(back)
28                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 28
                        lptr(back)=-la
                        back=la
c
27                      mini=min0(up,back)
c
                        nofs=1
                        if(up.eq.back) goto 29
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
c
29                      if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                        siteno=x*l+2+(l-1-z)*l*l
                        cluster(siteno)=mini
                        lptr(mini)=nofs
                else
                        c1=c1+1
                        siteno=x*l+2+(i-1-z)*l*l
                        cluster(siteno)=c1
                        lptr(c1)=1
                endif
                goto 10
        endif
c
        if(y.eq.l-1) then
                if(dia(x,l-1,z+1,3).ne.0.0  or. dia(x,l-1,z,1).ne.0.0
        1       .or. dia(x-1,l-1,z,2).ne.0.0 .or. dia(x,0,z,1).ne.0.0)
        2       then
c
c       Search the top connection
c
                        if(dia(x,l-1,z+1,3).eq.0.0) then
                                up=maxi
                                goto 30
                        else
                                if(z.eq.l-1) then
                                        siteup=1
                                else
                                        siteup=x*l+l+1+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
```

```
c
                        if(lptr(up).gt.0) goto 30
                        ms=lptr(up)
31                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 31
                        lptr(up)=-la
                        up=la
c
c       Search the left connection
c
30                      if(dia(x,l-1,z,1).eq.0.0) then
                                left=maxi
                                goto 32
                        else
                                sitelef=x*l+l+(l-1-z)*l*l
                                left=cluster(sitelef)
                        endif
c
                        if(lptr(left).gt.0) goto 32
                        ms=lptr(left)
33                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 33
                        lptr(left)=-la
                        left=la
c
c       Search the back connection
c
32                      if(dia(x-1,l-1,z,2).eq.0.0) then
                                back=maxi
                                goto 34
                        else
                                sitebck=(x-1)*l+l+1+(l-1-z)*l*l
                                back=cluster(sitebck)
                        endif
c
                        if(lptr(back).gt.0) goto 34
                        ms=lptr(back)
35                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 35
                        lptr(back)=-la
                        back=la
c
c       Search the right connection
c
34                      if(dia(x,0,z,1).eq.0.0) then
                                right=maxi
                                goto 36
                        else
                                siterit=x*l+2+(l-1-z)*l*l
                                right=cluster(siterit)
                        endif
c
                        if(lptr(right).gt.0) goto 36
                        ms=lptr(right)
37                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 37
                        lptr(right)=-la
                        right=la
c
36                      mini=min0(up,left,back,right)
```

```
c
                nofs=1
                if(up.eq.left .and. left.eq.back .and.
     1     back.eq.right) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                elseif(up.eq.left .and. left.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(right.ne.maxi) then
                                nofs=nofs+lptr(right)
                                lptr(right)=-mini
                        endif
                elseif(left.eq.back .and. back.eq.right) then
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                elseif(up.eq.back .and. back.eq.right) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                elseif(up.eq.left .and. left.eq.right) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                elseif(up.eq.left .and. back.eq.right) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                elseif(up.eq.right .and. left.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                elseif(up.eq.back .and. left.eq.right) then
                        if(up.ne.maxi) then
```

```
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.left) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(left.eq.back) then
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(back.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
```

```
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                elseif(left.eq.right) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                else
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                        if(right.ne.maxi) then
                                nofs=nofs+lptr(right)
                                lptr(right)=-mini
                        endif
                endif
                siteno=x*l+l+1+(l-1-z)*l*i
                cluster(siteno)=mini
                lptr(mini)=nofs
        else
                c1=c1+1
                siteno=x*l+l+1+(l-1-z)*l*l
                cluster(siteno)=c1
                lptr(c1)=1
        endif
        goto 10
endif
c
101     if(x.eq.l-1 .and. y.eq.0) then
                if(dia(l-1,0,z+1,3).ne.0.0 .or. dia(l-1,0,z,2).ne.0.0
        1       .or. dia(l-2,0,z,2).ne.0.0) then
c
c       Search the top connection
c
                        if(dia(l-1,0,z+1,3).eq.0.0) then
                                up=maxi
                                goto 39
                        else
                                if(z.eq.l-1) then
                                        siteup=1
                                else
```

```
                                        siteup=(l-1)*l+2+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
c
                        if(lptr(up).gt.0) goto 39
                        ms=lptr(up)
40                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 40
                        lptr(up)=-la
                        up=la
c
c       Search the front connection
c
39                      if(dia(l-1,0,z,2).eq.0.0) then
                                front=maxi
                                goto 41
                        else
                                sitefrt=2+(l-1-z)*l*l
                                front=cluster(sitefrt)
                        endif
c
                        if(lptr(front).gt.0) goto 41
                        ms=lptr(front)
42                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 42
                        lptr(front)=-la
                        front=la
c
c       Search the back connection
c
41                      if(dia(l-2,0,z,2).eq.0.0) then
                                back=maxi
                                goto 43
                        else
                                sitebck=(l-2)*l+2+(l-1-z)*l*l
                                back=cluster(sitebck)
                        endif
c
                        if(lptr(back).gt.0) goto 43
                        ms=lptr(back)
44                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 44
                        lptr(back)=-la
                        back=la
c
43                      mini=min0(up,front,back)
c
                        nofs=1
                        if(up.eq.front .and. front.eq.back) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                        elseif(up.eq.front) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(back.ne.maxi) then
                                        nofs=nofs+lptr(back)
```

```
                                lptr(back)=-mini
                        endif
                elseif(up.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                elseif(front.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                else
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                endif
                siteno=(l-1)*l+2+(l-1-z)*l*l
                cluster(siteno)=mini
                lptr(mini)=nofs
        else
                c1=c1+1
                siteno=(l-1)*l+2+(l-1-z)*l*l
                cluster(siteno)=c1
                lptr(c1)=1
        endif
        goto 10
endif
c
        if(x.eq.l-1) then
                if(dia(l-1,y,z+1,3).ne.0.0 .or. dia(l-1,y,z,1).ne.0.0
1               .or. dia(l-1,y,z,2).ne.0.0 .or. dia(l-2,y,z,2).ne.0.0)
2               then
c
c       Search the top connection
c
                        if(dia(l-1,y,z+1,3).eq.0.0) then
                                up=maxi
                                goto 46
                        else
                                if(z.eq.l-1) then
                                        siteup=1
                                else
                                        siteup=(l-1)*l+y+2+(l-z-2)*l*l
                                endif
                                up=cluster(siteup)
                        endif
```

```
c
                        if(lptr(up).gt.0) goto 46
                        ms=lptr(up)
47                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 47
                        lptr(up)=-la
                        up=la
c
c       Search the left connection
c
46                      if(dia(l-1,y,z,1).eq.0.0) then
                                left=maxi
                                goto 48
                        else
                                sitelef=(l-1)*l+y+1+(l-1-z)*l*l
                                left=cluster(sitelef)
                        endif
c
                        if(lptr(left).gt.0) goto 48
                        ms=lptr(left)
49                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 49
                        lptr(left)=-la
                        left=la
c
c       Search the front connection
c
48                      if(dia(l-1,y,z,2).eq.0.0) then
                                front=maxi
                                goto 50
                        else
                                sitefrt=y+2+(l-1-z)*l*l
                                front=cluster(sitefrt)
                        endif
c
                        if(lptr(front).gt.0) goto 50
                        ms=lptr(front)
51                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 51
                        lptr(front)=-la
                        front=la
c
c       Search the back connection
c
50                      if(dia(l-2,y,z,2).eq.0.0) then
                                back=maxi
                                goto 52
                        else
                                sitebck=(l-2)*l+y+2+(l-1-z)*l*l
                                back=cluster(sitebck)
                        endif
c
                        if(lptr(back).gt.0) goto 52
                        ms=lptr(back)
53                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 53
                        lptr(back)=-la
                        back=la
c
52                      mini=min0(up,left,front,back)
```

```
c
                nofs=1
                if(up.eq.left .and. left.eq.front .and.
     1      front.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                elseif(up.eq.left .and. left.eq.front) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                elseif(left.eq.front .and. front.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                elseif(up.eq.front .and. front.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                elseif(up.eq.left .and. left.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                elseif(up.eq.left .and. front.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                elseif(left.eq.front .and. up.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                elseif(up.eq.front .and. left.eq.back) then
                        if(up.ne.maxi) then
```

```
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.left) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(left.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
```

```
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                elseif(left.eq.back) then
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                else
                        if(up.ne.maxi) then
                                nofs=nofs+lptr(up)
                                lptr(up)=-mini
                        endif
                        if(left.ne.maxi)then
                                nofs=nofs+lptr(left)
                                lptr(left)=-mini
                        endif
                        if(front.ne.maxi) then
                                nofs=nofs+lptr(front)
                                lptr(front)=-mini
                        endif
                        if(back.ne.maxi) then
                                nofs=nofs+lptr(back)
                                lptr(back)=-mini
                        endif
                endif
                siteno=(l-1)*l+y+2+(l-1-z)*l*l
                cluster(siteno)=mini
                lptr(mini)=nofs
        else
                c1=c1+1
                siteno=(l-1)*l+y+2+(l-1-z)*l*l
                cluster(siteno)=c1
                lptr(c1)=1
        endif
        goto 10
endif
c
103     if(x.eq.l-1 .and. y.eq.l-1) then
        if(dia(l-1,l-1,z+1,3).ne.0.0 .or. dia(l-1,l-1,z,1).ne.0.0
1       .or. dia(l-1,l-1,z,2).ne.0.0 .or. dia(l-2,l-1,z,2).ne.0.0
2       .or. dia(l-1,0,z,1).ne.0.0) then
c
c       Search the top connection
c
                if(dia(l-1,l-1,z+1,3).eq.0.0) then
                        up=maxi
                        goto 55
                else
                        if(z.eq.l-1) then
                                siteup=1
```

```
                        else
                                siteup=(l-1)*l+l+1+(l-z-2)*l*l
                        endif
                        up=cluster(siteup)
                endif
c
                if(lptr(up).gt.0) goto 55
                ms=lptr(up)
56              la=-ms
                ms=lptr(la)
                if(ms.lt.0) goto 56
                lptr(up)=-la
                up=la
c
c       Search the left connection
c
55              if(dia(l-1,l-1,z,1).eq.0.0) then
                        left=maxi
                        goto 57
                else
                        sitelef=(l-1)*l+l+(l-1-z)*l*l
                        left=cluster(sitelef)
                endif
c
                if(lptr(left).gt.0) goto 57
                ms=lptr(left)
58              la=-ms
                ms=lptr(la)
                if(ms.lt.0) goto 58
                lptr(left)=-la
                left=la
c
c       Search the front connection
c
57              if(dia(l-1,l-1,z,2).eq.0.0) then
                        front=maxi
                        goto 59
                else
                        sitefrt=l+1+(l-1-z)*l*l
                        front=cluster(sitefrt)
                endif
c
                if(lptr(front).gt.0) goto 59
                ms=lptr(front)
60              la=-ms
                ms=lptr(la)
                if(ms.lt.0) goto 60
                lptr(front)=-la
                front=la
c
c       Search the back connection
c
59              if(dia(l-2,l-1,z,2).eq.0.0) then
                        back=maxi
                        goto 61
                else
                        sitebck=(l-2)*l+l+1+(l-1-z)*l*l
                        back=cluster(sitebck)
                endif
c
                if(lptr(back).gt.0) goto 61
                ms=lptr(back)
62              la=-ms
                ms=lptr(la)
```

```
                        if(ms.lt.0) goto 62
                        lptr(back)=-la
                        back=la
c
c       Search the right connection
c
61                      if(dia(l-1,0,z,1).eq.0.0) then
                                right=maxi
                                goto 63
                        else
                                siterit=(l-1)*l+2+(l-1-z)*l*l
                                right=cluster(siterit)
                        endif
c
                        if(lptr(right).gt.0) goto 63
                        ms=lptr(right)
64                      la=-ms
                        ms=lptr(la)
                        if(ms.lt.0) goto 64
                        lptr(right)=-la
                        right=la
c
63                      mini=min0(up,left,front,back,right)
c
                        nofs=1
                        if(up.eq.left .and. left.eq.front .and.
     1          front.eq.back .and. back.eq.right) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                        elseif(up.eq.left .and. left.eq.right .and.
     1          right.eq.front) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(back.ne.maxi) then
                                        nofs=nofs+lptr(back)
                                        lptr(back)=-mini
                                endif
                        elseif(left.eq.right .and. right.eq.front .and.
     1          front.eq.back) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                        elseif(up.eq.right .and. right.eq.front .and.
     1          front.eq.back) then
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                        elseif(up.eq.left .and. left.eq.front .and.
     1          front.eq.back) then
                                if(up.ne.maxi) then
```

```
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(right.ne.maxi) then
                        nofs=nofs+lptr(right)
                        lptr(right)=-mini
                    endif
                elseif(up.eq.left .and. left.eq.right .and.
     1          right.eq.back) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(front.ne.maxi) then
                        nofs=nofs+lptr(front)
                        lptr(front)=-mini
                    endif
                elseif(up.eq.left .and. left.eq.right .and.
     1          front.eq.back) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(back.ne.maxi) then
                        nofs=nofs+lptr(back)
                        lptr(back)=-mini
                    endif
                elseif(up.eq.left .and. left.eq.front .and.
     1          right.eq.back) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(right.ne.maxi) then
                        nofs=nofs+lptr(right)
                        lptr(right)=-mini
                    endif
                elseif(up.eq.left .and. left.eq.back .and.
     1          right.eq.front) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(right.ne.maxi) then
                        nofs=nofs+lptr(right)
                        lptr(right)=-mini
                    endif
                elseif(up.eq.right .and. right.eq.front .and.
     1          left.eq.back) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(left.ne.maxi) then
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                    endif
                elseif(up.eq.front .and. front.eq.back .and.
     1          left.eq.right) then
                    if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                    endif
                    if(left.ne.maxi) then
```

```
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                endif
        elseif(up.eq.right .and. right.eq.back .and.
1       left.eq.front) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(left.ne.maxi) then
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                endif
        elseif(left.eq.right .and. right.eq.front .and.
1       up.eq.back) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(left.ne.maxi) then
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                endif
        elseif(left.eq.right .and. right.eq.back .and.
1       up.eq.front) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(left.ne.maxi) then
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                endif
        elseif(left.eq.front .and. front.eq.back .and.
1       up.eq.right) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(left.ne.maxi) then
                        nofs=nofs+lptr(left)
                        lptr(left)=-mini
                endif
        elseif(right.eq.front .and. front.eq.back .and.
1       up.eq.left) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(right.ne.maxi) then
                        nofs=nofs+lptr(right)
                        lptr(right)=-mini
                endif
        elseif(up.eq.left .and. left.eq.right) then
                if(up.ne.maxi) then
                        nofs=nofs+lptr(up)
                        lptr(up)=-mini
                endif
                if(front.ne.maxi) then
                        nofs=nofs+lptr(front)
                        lptr(front)=-mini
                endif
                if(back.ne.maxi) then
                        nofs=nofs+lptr(back)
```

```
                lptr(back)=-mini
        endif
elseif(up.eq.left .and. left.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.left .and. left.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(left.eq.right .and. right.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(left.eq.right .and. right.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(left.eq.front .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
```

```
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.right .and. right.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.right .and. right.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(right.eq.front .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.front .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.left .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
```

```
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.left .and. right.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(up.eq.left .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.right .and. left.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi)then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.right .and. left.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(up.eq.right .and. front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
```

```
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.front .and. left.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.front .and. left.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.front .and. right.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.back .and. left.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(up.eq.back .and. left.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
```

```
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
elseif(up.eq.back .and. right.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
elseif(up.eq.left) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
```

```
                lptr(right)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(up.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(left.eq.right) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(left.eq.f
        if(up.ne.m      n
                no      s+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(left.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
```

```
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                iptr(right)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(right.eq.front) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
elseif(right.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(front.ne.maxi) then
                nofs=nofs+lptr(front)
                lptr(front)=-mini
        endif
elseif(front.eq.back) then
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
                lptr(up)=-mini
        endif
        if(left.ne.maxi) then
                nofs=nofs+lptr(left)
                lptr(left)=-mini
        endif
        if(right.ne.maxi) then
                nofs=nofs+lptr(right)
                lptr(right)=-mini
        endif
        if(back.ne.maxi) then
                nofs=nofs+lptr(back)
                lptr(back)=-mini
        endif
else
        if(up.ne.maxi) then
                nofs=nofs+lptr(up)
```

```
                                        lptr(up)=-mini
                                    endif
                                    if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                    endif
                                    if(right.ne.maxi) then
                                        nofs=nofs+lptr(right)
                                        lptr(right)=-mini
                                    endif
                                    if(front.ne.maxi) then
                                        nofs=nofs+lptr(front)
                                        lptr(front)=-mini
                                    endif
                                    if(back.ne.maxi) then
                                        nofs=nofs+lptr(back)
                                        lptr(back)=-mini
                                    endif
                                endif
                                siteno=(l-1)*l+l+1+(l-1-z)*l*l
                                cluster(siteno)=mini
                                lptr(mini)=nofs
                            else
                                c1=c1+1
                                siteno=(l-1)*l+l+1+(l-1-z)*l*l
                                cluster(siteno)=c1
                                lptr(c1)=1
                            endif
                            goto 10
                        endif
c
c
                            if(dia(x,y,z,1).ne.0.0 .or. dia(x,y,z+1,3).ne.0.0
                        1   .or. dia(x-1,y,z,2).ne.0.0) then
c
c       Search the top connection
c
                                if(dia(x,y,z+1,3).eq.0.0) then
                                    up=maxi
                                    goto 66
                                else
                                    if(z.eq.l-1) then
                                        siteup=1
                                    else
                                        siteup=x*l+y+2+(l-z-2)*l*l
                                    endif
                                    up=cluster(siteup)
                                endif
c
                                if(lptr(up).gt.0) goto 66
                                ms=lptr(up)
67                              la=-ms
                                ms=lptr(la)
                                if(ms.lt.0) goto 67
                                lptr(up)=-la
                                up=la
c
c       Search the left connection
c
66                              if(dia(x,y,z,1).eq.0.0) then
                                    left=maxi
                                    goto 68
                                else
                                    sitelef=x*l+y+1+(l-1-z)*l*l
```

```
                        left=cluster(sitelef)
                    endif
c
                    if(lptr(left).gt.0) goto 68
                    ms=lptr(left)
69                  la=-ms
                    ms=lptr(la)
                    if(ms.lt.0) goto 69
                    lptr(left)=-la
                    left=la
c
c       Finally the back connection
c
68                  if(dia(x-1,y,z,2).eq.0.0) then
                        back=maxi
                        goto 70
                    else
                        sitebck=(x-1)*l+y+2+(l-1-z)*l*l
                        back=cluster(sitebck)
                    endif
c
                    if(lptr(back).gt.0) goto 70
                    ms=lptr(back)
71                  la=-ms
                    ms=lptr(la)
                    if(ms.lt.0) goto 71
                    lptr(back)=-la
                    back=la
c
70                  mini=min0(up,left back)
c
                    nofs=1
                    if(up.eq.left .and. left.eq.back) then
                        if(up.ne.maxi) then
                            nofs=nofs+lptr(up)
                            lptr(up)=-mini
                        endif
                    elseif(up.eq.left) then
                        if(up.ne.maxi) then
                            nofs=nofs+lptr(up)
                            lptr(up)=-mini
                        endif
                        if(back.ne.maxi) then
                            nofs=nofs+lptr(back)
                            lptr(back)=-mini
                        endif
                    elseif(up.eq.back) then
                        if(up.ne.maxi) then
                            nofs=nofs+lptr(up)
                            lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                            nofs=nofs+lptr(left)
                            lptr(left)=-mini
                        endif
                    elseif(left.eq.back) then
                        if(up.ne.maxi) then
                            nofs=nofs+lptr(up)
                            lptr(up)=-mini
                        endif
                        if(left.ne.maxi) then
                            nofs=nofs+lptr(left)
                            lptr(left)=-mini
                        endif
```

```
                        else
                                if(up.ne.maxi) then
                                        nofs=nofs+lptr(up)
                                        lptr(up)=-mini
                                endif
                                if(left.ne.maxi) then
                                        nofs=nofs+lptr(left)
                                        lptr(left)=-mini
                                endif
                                if(back.ne.maxi) then
                                        nofs=nofs+lptr(back)
                                        lptr(back)=-mini
                                endif
                        endif
                        siteno=x*l+y+2+(l-1-z)*l*l
                        cluster(siteno)=mini
                        lptr(mini)=nofs
                else
                        c1=c1+1
                        siteno=x*l+y+2+(l-1-z)*l*l
                        cluster(siteno)=c1
                        lptr(c1)=1
                endif
10      continue
c
        do 2 i=1,(l-1)*l*l+2
                if(cluster(i).eq.maxi) then
                        siteocc(i)=i
                else
                        j=cluster(i)
                        if(lptr(j).gt.0 .and. j.ne.1) then
                                siteocc(i)=i
                                goto 2
                        endif
                        if(lptr(j).lt.0) then
                                ms=lptr(j)
3                               la=-ms
                                ms=lptr(la)
                                if(ms.lt.0) goto 3
                                if(la.eq.1) then
                                        siteocc(i)=0
                                else
                                        siteocc(i)=i
                                endif
                        else
                                siteocc(i)=0
                        endif
                endif
2       continue
c
        return
        end
```

Appendix A.2 Mercury Intrusion Simulation

```
      program mercury
c
c     This program simulates mercury porosimetry experiments on
c     a network generated from lattice.f
c
      integer x,y,z,orien,l
      real critdia,press,cumvol,intvol,k
      parameter (l=25, pie=3.141592654, k=214.3)
      real dia(0:l,0:l,0:l,3)
      integer fill(0:l,0:l,0:l,3)
      character*6 pfile
c
c
      data dia/52728*0.0/
      data fill/52728*0/cumvol,intvol/2*0.0/
c
      open (8, file="merv.d",status="new")
      call lattice(dia)
c
c     The program here simulates intrusion from critdia from 20
c     to 10 microns, but a second range from 9.5 to 0.007 microns
c     in increments of -0.001 is also to be used
c
      do 200 critdia = 20.0, 10.0, -1.0
c
c     Scan the surface pores
c
            do 230 z=1,l,l-1
                  do 230 x=0,l-1
                        do 230 y=1,l
                              do 230 orien=1,3
c
            if (dia(x,y,z,orien).ge.critdia .and. fill(x,y,z,orien).eq.0)
     1            then
                  fill(x,y,z,orien)=1
                  intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                  cumvol=cumvol+intvol
                  intvol=0.0
            endif
c
230         continue
c
c     Scan for interstitial pores to provide mercury meniscii
c
      if (critdia .ge. 1.0) then
            do 244 z=l,1,-1
                  do 244 x=0,l-1
                        do 244 y=1,l
                              do 244 orien=1,3
            if (fill(x,y,z,orien).eq.0 .and. dia(x,y,z,orien).ge.critdia)
     1      then
                  fill(x,y,z,orien) = 1
                  intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                  cumvol=cumvol+intvol
                  intvol=0.0
            endif
244         continue
      endif
c
c     Scan the internal pores individually to intrude mercury
c
```

```
      do 240 z=l-1,2,-1
            do 240 x=0,l-1
                  do 240 y=1,l
                        do 250 orien =1,3
c
            if ((dia(x,y,z,orien).ge.critdia).and.fill(x,y,z,orien).eq.0)
     1      then
                  if (orien .eq. 1) then
                  if(x.eq.0) then
                        if (fill(x,y-1,z,1).eq.1 .or.
     2                     fill(x,y-1,z,2).eq.1 .or.
     3                     fill(x,y-1,z,3).eq.1 .or.
     4                     fill(l-1,y,z,2).eq.1 .or.
     5                     fill(x,y,z+1,3).eq.1 .or.
     6                     fill(x,y-1,z+1,3).eq.1 .or.
     7                     fill(l-1,y-1,z,2).eq.1) then
c
                        fill(x,y,z,orien) = 1
                        intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                        endif
                  else
                        if (fill(x,y-1,z,1).eq.1 .or.
     2                     fill(x,y-1,z,2).eq.1 .or.
     3                     fill(x,y-1,z,3).eq.1 .or.
     4                     fill(x-1,y,z,2).eq.1 .or.
     5                     fill(x,y,z+1,3).eq.1 .or.
     6                     fill(x,y-1,z+1,3).eq.1 .or.
     7                     fill(x-1,y-1,z,2).eq.1) then
c
                        fill(x,y,z,orien) = 1
                        intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
c
                        endif
                  endif
                  elseif (orien .eq. 2) then
                  if(x.eq.0) then
                        if (fill(x,y,z,1).eq.1 .or.
     2                     fill(x,y,z+1,3).eq.1 .or.
     3                     fill(l-1,y,z,2).eq.1 .or.
     4                     fill(x+1,y,z+1,3).eq.1) then
                        fill(x,y,z,orien) = 1
                        intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                        endif
                  else
                        if (fill(x,y,z,1).eq.1 .or.
     2                     fill(x,y,z+1,3).eq.1 .or.
     3                     fill(x-1,y,z,2).eq.1 .or.
     4                     fill(x+1,y,z+1,3).eq.1) then
c
                        fill(x,y,z,orien) = 1
                        intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
c
                        endif
                  endif
                  elseif (orien .eq. 3) then
                  if(x.eq.0) then
```

```
                                    if (fill(x,y,z,1).eq.1 .or.
     2                                 fill(x,y,z,2).eq.1 .or.
     3                                 fill(x,y,z+1,3).eq.1 .or.
     4                                 fill(l-1,y,z,2).eq.1) then
c
                                    fill(x,y,z,orien)=1
                                    intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
                                    endif
                          else
                                    if (fill(x,y,z,1).eq.1 .or.
     2                                 fill(x,y,z,2).eq.1 .or.
     3                                 fill(x,y,z+1,3).eq.1 .or.
     4                                 fill(x-1,y,z,2).eq.1) then
c
                                    fill(x,y,z,orien)=1
                                    intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
c
                                    endif
                          endif
                          endif
c
               endif
250                                 continue
c         rescan the neighbouring pores to fill them if possible
c
                          if (dia(x,y,z,1).ge.critdia .and. fill(x,y,z,1).eq.1)
     1                    then
                                    if (dia(x,y,z,2).ge.critdia .and.
     1                                   fill(x,y,z,2).eq.0) then
                                    fill(x,y,z,2) = 1
                                    intvol=(pie*dia(x,y,z,2)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
                                    endif
                                    if (dia(x,y,z,3).ge.critdia .and.
     1                                   fill(x,y,z,3).eq.0) then
                                    fill(x,y,z,3) = 1
                                    intvol=(pie*dia(x,y,z,3)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
                                    endif
                                    if (x.eq.0.0) then
                                    if (dia(l-1,y,z,2).ge.critdia .and.
     1                                   fill(l-1,y,z,2).eq.0) then
                                    fill(l-1,y,z,2) = 1
                                    intvol=(pie*dia(l-1,y,z,2)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
                                    endif
                                    else
                                    if (dia(x-1,y,z,2).ge.critdia .and.
     1                                   fill(x-1,y,z,2).eq.0) then
                                    fill(x-1,y,z,2) = 1
                                    intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                                    cumvol=cumvol+intvol
                                    intvol=0.0
                                    endif
                                    endif
                                    if (dia(x,y,z+1,3).ge.critdia .and.
     1                                   fill(x,y,z+1,3).eq.0) then
```

```
                        fill(x,y,z+1,3) = 1
                        intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (dia(x,y-1,z,1).ge.critdia .and.
     1                  fill(x,y-1,z,1).eq.0) then
                        fill(x,y-1,z,1) = 1
                        intvol=(pie*dia(x,y-1,z,1)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (dia(x,y-1,z,2).ge.critdia .and.
     1                  fill(x,y-1,z,2).eq.0) then
                        fill(x,y-1,z,2) = 1
                        intvol=(pie*dia(x,y-1,z,2)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (dia(x,y-1,z,3).ge.critdia .and.
     1                  fill(x,y-1,z,3).eq.0) then
                        fill(x,y-1,z,3) = 1
                        intvol=(pie*dia(x,y-1,z,3)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (dia(x,y-1,z+1,3).ge.critdia .and.
     1                  fill(x,y-1,z+1,3).eq.0) then
                        fill(x,y-1,z+1,3) = 1
                        intvol=(pie*dia(x,y-1,z+1,3)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (x.eq.0) then
                    if (dia(l-1,y-1,z,2).ge.critdia .and.
     1                  fill(l-1,y-1,z,2).eq.0) then
                        fill(l-1,y-1,z,2) = 1
                        intvol=(pie*dia(l-1,y-1,z,2)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    else
                    if (dia(x-1,y-1,z,2).ge.critdia .and.
     1                  fill(x-1,y-1,z,2).eq.0) then
                        fill(x-1,y-1,z,2) = 1
                        intvol=(pie*dia(x-1,y-1,z,2)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    endif
                elseif (dia(x,y,z,2).ge.critdia .and. fill(x,y,z,2)
     1              .eq.1) then
                    if (dia(x,y,z,1).ge.critdia .and.
     1                  fill(x,y,z,1).eq.0) then
                        fill(x,y,z,1) = 1
                        intvol=(pie*dia(x,y,z,1)**3.0)/4.
                        cumvol=cumvol+intvol
                        intvol=0.0
                    endif
                    if (dia(x,y,z,3).ge.critdia .and.
     1                  fill(x,y,z,3).eq.0) then
                        fill(x,y,z,3) = 1
                        intvol=(pie*dia(x,y,z,3)**3.0)/4.
                        cumvol=cumvol+intvol
```

```
                    intvol=0.0
                endif
                if (x.eq.0) then
                if (dia(l-1,y,z,2).ge.critdia .and.
1                   fill(l-1,y,z,2).eq.0) then
                    fill(l-1,y,z,2) = 1
                    intvol=(pie*dia(l-1,y,z,2)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                else
                if (dia(x-1,y,z,2).ge.critdia .and.
1                   fill(x-1,y,z,2).eq.0) then
                    fill(x-1,y,z,2) = 1
                    intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                endif
                if (dia(x,y,z+1,3).ge.critdia .and.
1                   fill(x,y,z+1,3).eq.0) then
                    fill(x,y,z+1,3) = 1
                    intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                if (dia(x+1,y,z+1,3).ge.critdia .and.
1                   fill(x+1,y,z+1,3).eq.0) then
                    fill(x+1,y,z+1,3) = 1
                    intvol=(pie*dia(x+1,y,z+1,3)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
            elseif (dia(x,y,z,3).ge.critdia .and. fill(x,y,z,3)
1           .eq.1) then
                if (dia(x,y,z,1).ge.critdia .and.
1                   fill(x,y,z,1).eq.0) then
                    fill(x,y,z,1) = 1
                    intvol=(pie*dia(x,y,z,1)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                if (dia(x,y,z,2).ge.critdia .and.
1                   fill(x,y,z,2).eq.0) then
                    fill(x,y,z,2) = 1
                    intvol=(pie*dia(x,y,z,2)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                if (dia(x,y,z+1,3).ge.critdia .and.
1                   fill(x,y,z+1,3).eq.0) then
                    fill(x,y,z+1,3) = 1
                    intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
                if (x.eq.0) then
                if (dia(l-1,y,z,2).ge.critdia .and.
1                   fill(l-1,y,z,2).eq.0) then
                    fill(l-1,y,z,2) = 1
                    intvol=(pie*dia(l-1,y,z,2)**3.0)/4.
                    cumvol=cumvol+intvol
                    intvol=0.0
                endif
```

```
                                    else
                                    if (dia(x-1,y,z,2).ge.critdia .and.
     1                                      fill(x-1,y,z,2).eq.0) then
                                            fill(x-1,y,z,2) = 1
                                            intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                                            cumvol=cumvol+intvol
                                            intvol=0.0
                                    endif
                                    endif
                            endif
240      continue
c
         press = k/critdia
         write (8,*) press,cumvol
c
200      continue
c
         stop
         end
```

Appendix A.3 Protein Adsorption Simulation

```
      program adscap
c
c     This program the simulates accessible adsorption capacity of a
c     network generated from lattice.f at different protein sizes
c
      integer x,y,z,orien,l
      real critdia,ex,equi,cumacc,access
      parameter (l=25, pie=3.141592654)
      real dia(0:l,0:l,0:l,3)
      integer fill(0:l,0:l,0:l,3)
c
      data dia/52728*0.0/
      data fill/52728*0/
c
      open (6, file="adsorb.d",status="new")
      call lattice(dia)
c
      ex=2.0
c
      do 100 critdia = 0.2,0.002,-0.001
c
c     Scan the surface pores
c
      cumacc=0.0
c
      do 10 i=0,l
      do 10 j=0,l
      do 10 k=0,l
      do 10 orien=1,3
            fill(i,j,k,orien)=0
10    continue
c
            do 30 z=1,l,l-1
                  do 30 x=0,l-1
                        do 30 y=1,l
                              do 30 orien=1,3
c
            if (dia(x,y,z,orien).ge.critdia .and. fill(x,y,z,orien).eq.0)
     1            then
                  fill(x,y,z,orien)=1
                  equi=(1.0-critdia/dia(x,y,z,orien))**ex
                  access=equi*(pie*dia(x,y,z,orien)**2.0)
                  cumacc=cumacc+access
                  access=0.0
            endif
c
30          continue
c
c     Scan for interstitial pores
c
            do 400 z=l,1,-1
                  do 400 x=0,l-1
                        do 400 y=1,l
                              do 400 orien=1,3
            if (fill(x,y,z,orien).eq.0 .and. dia(x,y,z,orien).ge.1.0)
     1      then
                  fill(x,y,z,orien) = 1
                  equi=(1.0-critdia/dia(x,y,z,orien))**ex
                  access=equi*(pie*dia(x,y,z,orien)**2.0)
                  cumacc=cumacc+access
                  access=0.0
```

```
                endif
400             continue
c
c       Scan the intraparticle pores individually to assess accessibility
c
        do 40 z=l-1,2,-1
                do 40 x=0,l-1
                        do 40 y=1,l
                                do 50 orien =1,3
c
                if ((dia(x,y,z,orien).ge.critdia).and.fill(x,y,z,orien).eq.0)
        1       then
                        if (orien .eq. 1) then
                                if (fill(x,y-1,z,1).eq.1 .or.
        2                          fill(x,y-1,z,2).eq.1 .or.
        3                          fill(x,y-1,z,3).eq.1 .or.
        4                          fill(x-1,y,z,2).eq.1 .or.
        5                          fill(x,y,z+1,3).eq.1 .or.
        6                          fill(x,y-1,z+1,3).eq.1 .or.
        7                          fill(x-1,y-1,z,2).eq.1) then
c
                                fill(x,y,z,orien) = 1
                                equi=(1.0-critdia/dia(x,y,z,orien))**ex
                                access=equi*(pie*dia(x,y,z,orien)**2.0)
                                cumacc=cumacc+access
                                access=0.0
c
                                endif
                        elseif (orien .eq. 2) then
                                if (fill(x,y,z,1).eq.1 .or.
        2                          fill(x,y,z+1,3).eq.1 .or.
        3                          fill(x-1,y,z,2).eq.1 .or.
        4                          fill(x+1,y,z+1,3).eq.1) then
c
                                fill(x,y,z,orien) = 1
                                equi=(1.0-critdia/dia(x,y,z,orien))**ex
                                access=equi*(pie*dia(x,y,z,orien)**2.0)
                                cumacc=cumacc+access
                                access=0.0
c
                                endif
                        elseif (orien .eq. 3) then
                                if (fill(x,y,z,1).eq.1 .or.
        2                          fill(x,y,z,2).eq.1 .or.
        3                          fill(x,y,z+1,3).eq.1 .or.
        4                          fill(x-1,y,z,2).eq.1) then
c
                                fill(x,y,z,orien)=1
                                equi=(1.0-critdia/dia(x,y,z,orien))**ex
                                access=equi*(pie*dia(x,y,z,orien)**2.0)
                                cumacc=cumacc+access
                                access=0.0
c
                                endif
                        endif
c
                endif
50                              continue
c       Rescan the neighbouring pores to assess accessibility
c
                        if (dia(x,y,z,1).ge.critdia .and. fill(x,y,z,1).eq.1)
        1               then
                                if (dia(x,y,z,2).ge.critdia .and.
        1                               fill(x,y,z,2).eq.0) then
```

```
                        fill(x,y,z,2) = 1
                        equi=(1.0-critdia/dia(x,y,z,2))**ex
                        access=equi*(pie*dia(x,y,z,2)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y,z,3).ge.critdia .and.
1                       fill(x,y,z,3).eq.0) then
                        fill(x,y,z,3) = 1
                        equi=(1.0-critdia/dia(x,y,z,3))**ex
                        access=equi*(pie*dia(x,y,z,3)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x-1,y,z,2).ge.critdia .and.
1                       fill(x-1,y,z,2).eq.0) then
                        fill(x-1,y,z,2) = 1
                        equi=(1.0-critdia/dia(x-1,y,z,2))**ex
                        access=equi*(pie*dia(x-1,y,z,2)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y,z+1,3).ge.critdia .and.
1                       fill(x,y,z+1,3).eq.0) then
                        fill(x,y,z+1,3) = 1
                        equi=(1.0-critdia/dia(x,y,z+1,3))**ex
                        access=equi*(pie*dia(x,y,z+1,3)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y-1,z,1).ge.critdia .and.
1                       fill(x,y-1,z,1).eq.0) then
                        fill(x,y-1,z,1) = 1
                        equi=(1.0-critdia/dia(x,y-1,z,1))**ex
                        access=equi*(pie*dia(x,y-1,z,1)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y-1,z,2).ge.critdia .and.
1                       fill(x,y-1,z,2).eq.0) then
                        fill(x,y-1,z,2) = 1
                        equi=(1.0-critdia/dia(x,y-1,z,2))**ex
                        access=equi*(pie*dia(x,y-1,z,2)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y-1,z,3).ge.critdia .and.
1                       fill(x,y-1,z,3).eq.0) then
                        fill(x,y-1,z,3) = 1
                        equi=(1.0-critdia/dia(x,y-1,z,3))**ex
                        access=equi*(pie*dia(x,y-1,z,3)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x,y-1,z+1,3).ge.critdia .and.
1                       fill(x,y-1,z+1,3).eq.0) then
                        fill(x,y-1,z+1,3) = 1
                    equi=(1.0-critdia/dia(x,y-1,z+1,3))**ex
                    access=equi*(pie*dia(x,y-1,z+1,3)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    elseif (dia(x-1,y-1,z,2).ge.critdia .and.
1                       fill(x-1,y-1,z,2).eq.0) then
                        fill(x-1,y-1,z,2) = 1
                    equi=(1.0-critdia/dia(x-1,y-1,z,2))**ex
                    access=equi*(pie*dia(x-1,y-1,z,2)**2.0)
                        cumacc=cumacc+access
                        access=0.0
                    endif
                elseif (dia(x,y,z,2).ge.critdia .and. fill(x,y,z,2)
1               .eq.1) then
```

```
                    if (dia(x,y,z,1).ge.critdia .and.
1                           fill(x,y,z,1).eq.0) then
                            fill(x,y,z,1) = 1
                    equi=(1.0-critdia/dia(x,y,z,1))**ex
                    access=equi*(pie*dia(x,y,z,1)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x,y,z,3).ge.critdia .and.
1                           fill(x,y,z,3).eq.0) then
                            fill(x,y,z,3) = 1
                    equi=(1.0-critdia/dia(x,y,z,3))**ex
                    access=equi*(pie*dia(x,y,z,3)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x-1,y,z,2).ge.critdia .and.
1                           fill(x-1,y,z,2).eq.0) then
                            fill(x-1,y,z,2) = 1
                    equi=(1.0-critdia/dia(x-1,y,z,2))**ex
                    access=equi*(pie*dia(x-1,y,z,2)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x,y,z+1,3).ge.critdia .and.
1                           fill(x,y,z+1,3).eq.0) then
                            fill(x,y,z+1,3) = 1
                    equi=(1.0-critdia/dia(x,y,z+1,3))**ex
                    access=equi*(pie*dia(x,y,z+1,3)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x+1,y,z+1,3).ge.critdia .and.
1                           fill(x+1,y,z+1,3).eq.0) then
                            fill(x+1,y,z+1,3) = 1
                    equi=(1.0-critdia/dia(x+1,y,z+1,3))**ex
                    access=equi*(pie*dia(x+1,y,z+1,3)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    endif
            elseif (dia(x,y,z,3).ge.critdia .and. fill(x,y,z,3)
1           .eq.1) then
                    if (dia(x,y,z,1).ge.critdia .and.
1                           fill(x,y,z,1).eq.0) then
                            fill(x,y,z,1) = 1
                    equi=(1.0-critdia/dia(x,y,z,1))**ex
                    access=equi*(pie*dia(x,y,z,1)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x,y,z,2).ge.critdia .and.
1                           fill(x,y,z,2).eq.0) then
                            fill(x,y,z,2) = 1
                    equi=(1.0-critdia/dia(x,y,z,2))**ex
                    access=equi*(pie*dia(x,y,z,2)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x,y,z+1,3).ge.critdia .and.
1                           fill(x,y,z+1,3).eq.0) then
                            fill(x,y,z+1,3) = 1
                    equi=(1.0-critdia/dia(x,y,z+1,3))**ex
                    access=equi*(pie*dia(x,y,z+1,3)**2.0)
                            cumacc=cumacc+access
                            access=0.0
                    elseif (dia(x-1,y,z,2).ge.critdia .and.
1                           fill(x-1,y,z,2).eq.0) then
                            fill(x-1,y,z,2) = 1
                    equi=(1.0-critdia/dia(x-1,y,z,2))**ex
                    access=equi*(pie*dia(x-1,y,z,2)**2.0)
```

```
                                        cumacc=cumacc+access
                                        access=0.0
                                    endif
                                endif
40      continue
c
        write (6,*) critdia,cumacc
c
100     continue
500     format(a6)
c
        stop
        end
```

Appendix A.4 Size Exclusion Chromatography Simulation

```
      program gel
c
c     This program simulates the partition coefficient for PS
c     at different gyration diameters for lattice.f generation
c
      integer x,y,z,orien,l
      real bigr,crit,lambda,len,keq,porevol,critdia,cumvol,intvol
      parameter (l=25, pie=3.141592654)
      real dia(0:l,0:l,0:l,3),alpha(52)
      integer fill(0:l,0:l,0:l,3)
      character*6 pfile
c
c
      data dia/52728*0.0/
      data fill/52728*0/porevol,cumvol,intvol/3*0.0/
c
      open (7, file="sec.d",status="new")
      open (6, file="broot.d",status="old")
c
      call lattice(dia)
      do 5 i=1,52
5           read(6,*) alpha(i)
c
      do 100 critdia = 0.2,0.002,-0.001
c
      crit=0.667*critdia
c
      do 10 i=0,l
      do 10 j=0,l
      do 10 k=0,l
      do 10 orien=1,3
            fill(i,j,k,orien)=0
10    continue
c
c     Scan for interstitial pores
c
            do 400 z=l,1,-1
                  do 400 x=0,l-1
                        do 400 y=1,l
                              do 400 orien=1,3
            if (dia(x,y,z,orien).ge.1.0) then
                  fill(x,y,z,orien) = 1
            endif
400         continue
c
c     Scan the surface pores
c
      cumvol=0.0
c
      z=l
                  do 30 x=0,l-1
                        do 30 y=1,l
                              do 30 orien=1,3
c
            if (dia(x,y,z,orien).gt.crit .and. fill(x,y,z,orien).eq.0)
     1      then
                  fill(x,y,z,orien)=1
                  intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                  call seglen(critdia,len)
                  lambda=critdia/dia(x,y,z,orien)
                  bigr=dia(x,y,z,orien)
```

```
                        call equi(alpha,lambda,len,bigr,critdia,keq)
                        intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
c
30              continue
c
c       Scan the internal pores
c
        do 40 z=l-1,1,-1
                do 40 x=0,l-1
                        do 40 y=1,l
                                do 50 orien =1,3
c
                if (dia(x,y,z,orien).gt.crit .and. fill(x,y,z,orien).eq.0)
        1       then
                        if (orien .eq. 1) then
                                if (fill(x,y-1,z,1).eq.1 .or.
        2                           fill(x,y-1,z,2).eq.1 .or.
        3                           fill(x,y-1,z,3).eq.1 .or.
        4                           fill(x-1,y,z,2).eq.1 .or.
        5                           fill(x,y,z+1,3).eq.1 .or.
        6                           fill(x,y-1,z+1,3).eq.1 .or.
        7                           fill(x-1,y-1,z,2).eq.1) then
c
                                fill(x,y,z,orien) = 1
                                intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,orien)
                                bigr=dia(x,y,z,orien)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                cumvol=cumvol+intvol
                                intvol=0.0
c
                                endif
                        elseif (orien .eq. 2) then
                                if (fill(x,y,z,1).eq.1 .or.
        2                           fill(x,y,z+1,3).eq.1 .or.
        3                           fill(x-1,y,z,2).eq.1 .or.
        4                           fill(x+1,y,z+1,3).eq.1) then
c
                                fill(x,y,z,orien) = 1
                                intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,orien)
                                bigr=dia(x,y,z,orien)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                cumvol=cumvol+intvol
                                intvol=0.0
c
                                endif
                        elseif (orien .eq. 3) then
                                if (fill(x,y,z,1).eq.1 .or.
        2                           fill(x,y,z,2).eq.1 .or.
        3                           fill(x,y,z+1,3).eq.1 .or.
        4                           fill(x-1,y,z,2).eq.1) then
c
                                fill(x,y,z,orien)=1
                                intvol=(pie*dia(x,y,z,orien)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,orien)
```

```
                                bigr=dia(x,y,z,orien)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                cumvol=cumvol+intvol
                                intvol=0.0
c
                                endif
                        endif
c
                endif
50                              continue
c
c       Rescan the neighbouring pores
c
                        if (fill(x,y,z,1).eq.1) then
                                if (dia(x,y,z,2).gt.crit .and. fill(x,y,z,2)
     1                          .eq.0) then
                                        fill(x,y,z,2) = 1
                                        intvol=(pie*dia(x,y,z,2)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,2)
                                bigr=dia(x,y,z,2)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x,y,z,3).gt.crit .and. fill(x,y,z,3)
     1                          .eq.0) then
                                        fill(x,y,z,3) = 1
                                        intvol=(pie*dia(x,y,z,3)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,3)
                                bigr=dia(x,y,z,3)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x-1,y,z,2).gt.crit .and.
      1                         fill(x-1,y,z,2).eq.0) then
                                        fill(x-1,y,z,2) = 1
                                        intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x-1,y,z,2)
                                bigr=dia(x-1,y,z,2)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x,y,z+1,3).gt.crit .and.
     1                          fill(x,y,z+1,3).eq.0) then
                                        fill(x,y,z+1,3) = 1
                                        intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z+1,3)
                                bigr=dia(x,y,z+1,3)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x,y-1,z,1).gt.crit .and.
```

```
1                   fill(x,y-1,z,1).eq.0) then
                            fill(x,y-1,z,1) = 1
                            intvol=(pie*dia(x,y-1,z,1)**3.0)/4.
                    call seglen(critdia,len)
                    lambda=critdia/dia(x,y-1,z,1)
                    bigr=dia(x,y-1,z,1)
                    call equi(alpha,lambda,len,bigr,critdia,keq)
                    intvol=keq*intvol
                            cumvol=cumvol+intvol
                            intvol=0.0
                    endif
                    if (dia(x,y-1,z,2).gt.crit .and.
1                   fill(x,y-1,z,2).eq.0) then
                            fill(x,y-1,z,2) = 1
                            intvol=(pie*dia(x,y-1,z,2)**3.0)/4.
                    call seglen(critdia,len)
                    lambda=critdia/dia(x,y-1,z,2)
                    bigr=dia(x,y-1,z,2)
                    call equi(alpha,lambda,len,bigr,critdia,keq)
                    intvol=keq*intvol
                            cumvol=cumvol+intvol
                            intvol=0.0
                    endif
                    if (dia(x,y-1,z,3).gt.crit .and.
1                   fill(x,y-1,z,3).eq.0) then
                            fill(x,y-1,z,3) = 1
                            intvol=(pie*dia(x,y-1,z,3)**3.0)/4.
                    call seglen(critdia,len)
                    lambda=critdia/dia(x,y-1,z,3)
                    bigr=dia(x,y-1,z,3)
                    call equi(alpha,lambda,len,bigr,critdia,keq)
                    intvol=keq*intvol
                            cumvol=cumvol+intvol
                            intvol=0.0
                    endif
                    if (dia(x,y-1,z+1,3).gt.crit .and.
1                   fill(x,y-1,z+1,3).eq.0) then
                            fill(x,y-1,z+1,3) = 1
                            intvol=(pie*dia(x,y-1,z+1,3)**3.0)/4.
                    call seglen(critdia,len)
                    lambda=critdia/dia(x,y-1,z+1,3)
                    bigr=dia(x,y-1,z+1,3)
                    call equi(alpha,lambda,len,bigr,critdia,keq)
                    intvol=keq*intvol
                            cumvol=cumvol+intvol
                            intvol=0.0
                    endif
                    if (dia(x-1,y-1,z,2).gt.crit .and.
1                   fill(x-1,y-1,z,2).eq.0) then
                            fill(x-1,y-1,z,2) = 1
                            intvol=(pie*dia(x-1,y-1,z,2)**3.0)/4.
                    call seglen(critdia,len)
                    lambda=critdia/dia(x-1,y-1,z,2)
                    bigr=dia(x-1,y-1,z,2)
                    call equi(alpha,lambda,len,bigr,critdia,keq)
                    intvol=keq*intvol
                            cumvol=cumvol+intvol
                            intvol=0.0
                    endif
            elseif (fill(x,y,z,2).eq.1) then
                    if (dia(x,y,z,1).gt.crit .and.
1                   fill(x,y,z,1).eq.0) then
                            fill(x,y,z,1) = 1
                            intvol=(pie*dia(x,y,z,1)**3.0)/4.
```

```
                call seglen(critdia,len)
                lambda=critdia/dia(x,y,z,1)
                bigr=dia(x,y,z,1)
                call equi(alpha,lambda,len,bigr,critdia,keq)
                intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
                if (dia(x,y,z,3).gt.crit .and.
1               fill(x,y,z,3).eq.0) then
                        fill(x,y,z,3) = 1
                        intvol=(pie*dia(x,y,z,3)**3.0)/4.
                call seglen(critdia,len)
                lambda=critdia/dia(x,y,z,3)
                bigr=dia(x,y,z,3)
                call equi(alpha,lambda,len,bigr,critdia,keq)
                intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
                if (dia(x-1,y,z,2).gt.crit .and.
1               fill(x-1,y,z,2).eq.0) then
                        fill(x-1,y,z,2) = 1
                        intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                call seglen(critdia,len)
                lambda=critdia/dia(x-1,y,z,2)
                bigr=dia(x-1,y,z,2)
                call equi(alpha,lambda,len,bigr,critdia,keq)
                intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
                if (dia(x,y,z+1,3).gt.crit .and.
1               fill(x,y,z+1,3).eq.0) then
                        fill(x,y,z+1,3) = 1
                        intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                call seglen(critdia,len)
                lambda=critdia/dia(x,y,z+1,3)
                bigr=dia(x,y,z+1,3)
                call equi(alpha,lambda,len,bigr,critdia,keq)
                intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
                if (dia(x+1,y,z+1,3).gt.crit .and.
1               fill(x+1,y,z+1,3).eq.0) then
                        fill(x+1,y,z+1,3) = 1
                        intvol=(pie*dia(x+1,y,z+1,3)**3.0)/4.
                call seglen(critdia,len)
                lambda=critdia/dia(x+1,y,z+1,3)
                bigr=dia(x+1,y,z+1,3)
                call equi(alpha,lambda,len,bigr,critdia,keq)
                intvol=keq*intvol
                        cumvol=cumvol+intvol
                        intvol=0.0
                endif
        elseif (fill(x,y,z,3).eq.1) then
                if (dia(x,y,z,1).gt.crit .and.
1               fill(x,y,z,1).eq.0) then
                        fill(x,y,z,1) = 1
                        intvol=(pie*dia(x,y,z,1)**3.0)/4.
                call seglen(critdia,len)
                lambda=critdia/dia(x,y,z,1)
                bigr=dia(x,y,z,1)
```

```
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x,y,z,2).gt.crit .and.
     1                          fill(x,y,z,2).eq.0) then
                                        fill(x,y,z,2) = 1
                                        intvol=(pie*dia(x,y,z,2)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z,2)
                                bigr=dia(x,y,z,2)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x,y,z+1,3).gt.crit .and.
     1                          fill(x,y,z+1,3).eq.0) then
                                        fill(x,y,z+1,3) = 1
                                        intvol=(pie*dia(x,y,z+1,3)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x,y,z+1,3)
                                bigr=dia(x,y,z+1,3)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                                if (dia(x-1,y,z,2).gt.crit .and.
     1                          fill(x-1,y,z,2).eq.0) then
                                        fill(x-1,y,z,2) = 1
                                        intvol=(pie*dia(x-1,y,z,2)**3.0)/4.
                                call seglen(critdia,len)
                                lambda=critdia/dia(x-1,y,z,2)
                                bigr=dia(x-1,y,z,2)
                                call equi(alpha,lambda,len,bigr,critdia,keq)
                                intvol=keq*intvol
                                        cumvol=cumvol+intvol
                                        intvol=0.0
                                endif
                        endif
40      continue
c
        write (7,*) critdia, cumvol
c
100     continue
500     format(a6)
c
        stop
        end



        subroutine seglen(cdia,len)
c
c       This subroutine computes the segment length of the polystyrene
c       molecule whose radius of gyration is critdia
c
        real cdia,len,ex,molwt,sn
c
        ex=1.7007
        molwt = (cdia*10000.0/0.2840)**ex
        sn = molwt/104.
```

```
      len = (0.5*cdia)*2.44949/sqrt(sn-1.0/sn)
c
      return
      end


      subroutine equi(alpha,lambda,len,bigr,critdia,keq)
c
c     This subroutine calculates the exclusion coefficient for the
c     polystyrene solute based on an equilibrium partitioning model
c     for flexible macromolecules
c
      real alpha(52), lambda,len,bigr,critdia,keq,kzero,lnk
c
      kzero = 0.0
      do 20 j=1,52
20          kzero=kzero+(1.0/alpha(j)**2.0)*exp(-1.0*alpha(j)**2.0
     1      *lambda**2.0)
c
      kzero = 4.0*kzero
c
      lnk = alog(kzero)+2.0*len/bigr*(0.49+1.09*lambda+1.79*lambda
     2      *lambda)
c
      keq = exp(lnk)
c
      return
      end
```

Appendix A.5 Medium Permeability Simulation

```
      program pressure
c
c     This program performs nodal pressure calculations for a
c     network generated from lattice.f
c
      integer x,y,z,orien,l,info,warn
      real big,superv,pie,mu,kon,const,const1,linvel,xarea,flrate
      integer siteino,sitelef,siterit,sitefrt,sitebck,sitetop,sitebtm
      integer ileft,iright,ifront,iback,itop,ibtm
      integer isite,siteno,warnlef,warnrit,warnfrt,warnbck,warntop
      real asum,delp
      integer m(15002),m1,n,nrhs,k,lda,ldb,kl,ku,warnbtm
      character*1 trans
      parameter (k1=1984,k2=15002,k3=1876,k4=k2-k1,k5=k3*k4)
c
c     the dimension of array a is specified as 3*l**2+1
c
      real a(k3,k4)
c
c     the dimension of ipiv is specified as 2+l**2*(l-1)
c
      integer ipiv(k4)
c
c     the dimension of array b in AX=b is 2+(l-1)*l**2
c
      real b(k4)
c
c     Superficial velocity (superv) is in cm/h
c     Viscosity (mu) is in g/cm-s
c
      parameter (l=25, pie=3.141592654, mu=0.01)
      real dia(0:l,0:l,0:l,3)
      data dia/52728*0.0/
      data b/k4*0.0/m/15002*0/
c
      data a/k5*0.0/siteino,sitelef,siterit,sitefrt,sitebck/5*0/
      data sitetop,sitebtm,ileft,iright,ifront,iback,itop,ibtm/8*0/
      data siteno,isite/2*0/
c
      open(8, file="output.try", status="old")
c
      const=pie/(203.4*mu)
      kl=l**2
      ku=l**2
      m1=2+(l-1)*l**2-k1
      n=m1
      trans='n'
      nrhs=1
      k=kl+ku+1
      lda=2*kl+ku+1
      ldb=n
c
      call lattice(dia,volcell,m)
c
c     Flowrate (flrate) is in (um)^3/s
c     Linear Velocity (linvel) is in um/s
c
      read(8,*) superv
      read(8,*) vollen, xarea
      linvel=superv*0.27777778
```

```
      flrate = linvel*xarea
      b(1)=flrate
      b(m1)=-1.0*flrate
c
c     Pressure equations formulated for the left and right periodic
c     surfaces
c
      do 100 y=0,l-1,l-1
      do 100 z=l-1,1,-1
      do 100 x=1,l-1
            siteino = l*x+y+2+(l-1-z)*l*l
            call correct(siteino,warn,m,k1)
            if (warn .eq.1) goto 100
            if (y.eq.0) then
                  sitelef = l*x+l+1+(l-1-z)*l*l
                  call correct(sitelef,warn,m,k1)
                  if (warn.eq.1) warnlef=1
                  siterit = l*x+3+(l-1-z)*l*l
                  call correct(siterit,warn,m,k1)
                  if (warn.eq.1) warnrit=1
            else
                  sitelef = l*x+l+(l-1-z)*l*l
                  call correct(sitelef,warn,m,k1)
                  if (warn.eq.1) warnlef=1
                  siterit = l*x+2+(l-1-z)*l*l
                  call correct(siterit,warn,m,k1)
                  if (warn.eq.1) warnrit=1
            endif
            if (x .eq. l-1) then
                  sitefrt = y+2+(l-1-z)*l*l
            else
                  sitefrt = (x+1)*l+y+2+(l-1-z)*l*l
            endif
            call correct (sitefrt,warn,m,k1)
            if (warn.eq.1) warnfrt=1
            sitebck = (x-1)*l+y+2+(l-1-z)*l*l
            call correct (sitebck,warn,m,k1)
            if (warn.eq.1) warnbck=1
            if (z.eq.l-1) then
                  sitetop = 1
                  sitebtm = l*x+y+2+l*l
                  call correct(sitebtm,warn,m,k1)
                  if (warn.eq.1) warnbtm=1
            elseif (z.eq.1) then
                  sitetop = l*x+y+2+(l-3)*l*l
                  call correct(sitetop,warn,m,k1)
                  if (warn.eq.1) warntop=1
                  sitebtm = 2+(l-1)*l**2-k1
            else
                  sitetop = l*x+y+2+(l-2-z)*l*l
                  call correct(sitetop,warn,m,k1)
                  if (warn.eq.1) warntop=1
                  sitebtm = l*x+y+2+(l-z)*l*l
                  call correct(sitebtm,warn,m,k1)
                  if (warn.eq.1) warnbtm=1
            endif
c
            asum = 0.0
            if (warnlef .ne.1) then
                  ileft = siteino-sitelef+k
                  a(ileft,sitelef)=-1.0*const*dia(x,y,z,1)**3.0
                  asum = asum-a(ileft,sitelef)
            else
                  warnlef=0
```

```
              endif
              if (warnrit .ne. 1) then
                      iright = siteino-siterit+k
                      if (y.eq.0) then
                              a(iright,siterit)=-1.0*const*
     3                        dia(x,1,z,1)**3.0
                      else
                              a(iright,siterit)=-1.0*const*
     4                        dia(x,0,z,1)**3.0
                      endif
                      asum = asum-a(iright,siterit)
              else
                      warnrit=0
              endif
              if (warnfrt .ne. 1) then
                      ifront = siteino-sitefrt+k
                      a(ifront,sitefrt)=-1.0*const*dia(x,y,z,2)**3.0
                      asum = asum-a(ifront,sitefrt)
              else
                      warnfrt=0
              endif
              if (warnbck .ne. 1) then
                      iback = siteino-sitebck+k
                      a(iback,sitebck)=-1.0*const*dia(x-1,y,z,2)**3.0
                      asum = asum-a(iback,sitebck)
              else
                      warnbck=0
              endif
              if (warntop .ne. 1) then
                      itop = siteino-sitetop+k
                      a(itop,sitetop)=-1.0*const*dia(x,y,z+1,3)**3.0
                      asum = asum-a(itop,sitetop)
              else
                      warntop=0
              endif
              if (warnbtm .ne. 1) then
                      ibtm = siteino-sitebtm+k
                      a(ibtm,sitebtm)=-1.0*const*dia(x,y,z,3)**3.0
                      asum = asum-a(ibtm,sitebtm)
              else
                      warnbtm=0
              endif
              a(k,siteino)=asum
  100 continue
c
c     Pressure equations formulated for the front and back periodic
c     surfaces
c
      do 200 x=0,l-1,l-1
      do 200 z=l-1,1,-1
      do 200 y=1,l-1
              siteino = l*x+y+2+(l-1-z)*l*l
              call correct(siteino,warn,m,k1)
              if (warn.eq.1) goto 200
              sitelef = l*x+y+1+(l-1-z)*l*l
              call correct(sitelef,warn,m,k1)
              if (warn.eq.1) warnlef=1
              if (y .eq. l-1) then
                      siterit = x*l+2+(l-1-z)*l*l
              else
                      siterit = x*l+y+3+(l-1-z)*l*l
              endif
              call correct(siterit,warn,m,k1)
              if (warn.eq.1) warnrit=1
```

```
      if (x.eq.l-1) then
            sitefrt = y+2+(l-1-z)*l*l
            call correct(sitefrt,warn,m,k1)
            if (warn.eq.1) warnfrt=1
            sitebck = (l-2)*l+y+2+(l-1-z)*l*l
            call correct(sitebck,warn,m,k1)
            if (warn.eq.1) warnbck=1
      else
            sitefrt = l+y+2+(l-1-z)*l*l
            call correct(sitefrt,warn,m,k1)
            if (warn..eq.1) warnfrt=1
            sitebck = l*(l-1)+y+2+(l-1-z)*l*l
            call correct(sitebck,warn,m,k1)
            if (warn.eq.1) warnbck=1
      endif
      if (z.eq.l-1) then
            sitetop = 1
            sitebtm = l*x+y+2+l*l
            call correct(sitebtm,warn,m,k1)
            if (warn.eq.1) warnbtm=1
      elseif (z.eq.1) then
            sitetop = l*x+y+2+(l-3)*l*l
            call correct(sitetop,warn,m,k1)
            if (warn.eq.1) warntop=1
            sitebtm = 2+(l-1)*l*l-k1
      else
            sitetop = l*x+y+2+(l-2-z)*l*l
            call correct(sitetop,warn,m,k1)
            if (warn.eq.1) warntop=1
            sitebtm = l*x+(y+2)+(l-z)*l*l
            call correct(sitebtm,warn,m,k1)
            if (warn.eq.1) warnbtm=1
      endif
      asum = 0.0
      if (warnlef .ne. 1) then
            ileft = siteino-sitelef+k
            a(ileft,sitelef)=-1.0*const*dia(x,y,z,1)**3.0
            asum = asum-a(ileft,sitelef)
      else
            warnlef=0
      endif
      if (warnrit .ne. 1) then
            iright = siteino-siterit+k
            a(iright,siterit)=-1.0*const*dia(x,y+1,z,1)**3.0
            asum = asum-a(iright,siterit)
      else
            warnrit=0
      endif
      if (warnfrt .ne. 1) then
            ifront = siteino-sitefrt+k
            a(ifront,sitefrt)=-1.0*const*dia(x,y,z,2)**3.0
            asum = asum-a(ifront,sitefrt)
      else
            warnfrt=0
      endif
      if (warnbck .ne.1) then
            iback = siteino-sitebck+k
            if (x.eq.0) then
                  a(iback,sitebck)=-1.0*const*
     1            dia(l-1,y,z,2)**3.0
            else
                  a(iback,sitebck)=-1.0*const*
     2            dia(l-2,y,z,2)**3.0
            endif
```

```
                asum = asum-a(iback,sitebck)
        else
                warnbck=0
        endif
        if (warntop .ne. 1) then
                itop = siteino-sitetop+k
                a(itop,sitetop)=-1.0*const*dia(x,y,z+1,3)**3.0
                asum = asum-a(itop,sitetop)
        else
                warntop=0
        endif
        if (warnbtm .ne. 1) then
                ibtm = siteino-sitebtm+k
                a(ibtm,sitebtm)=-1.0*const*dia(x,y,z,3)**3.0
                asum = asum-a(ibtm,sitebtm)
        else
                warnbtm=0
        endif
        a(k,siteino) = asum
200     continue
c
c       Pressure equations are formulated for the left,back spine
c
        x=0
        y=0
        do 300 z=1,l-1
                siteino = 2+(l-1-z)*l*l
                call correct(siteino,warn,m,k1)
                if (warn.eq.1) goto 300
                sitelef = l+1+(l-1-z)*l*l
                call correct(sitelef,warn,m,k1)
                if (warn.eq.1) warnlef=1
                siterit = 3+(l-1-z)*l*l
                call correct(siterit,warn,m,k1)
                if (warn.eq.1) warnrit=1
                sitefrt = l+2+(l-1-z)*l*l
                call correct(sitefrt,warn,m,k1)
                if (warn.eq.1) warnfrt=1
                sitebck = l*(l-1)+2+(l-1-z)*l*l
                call correct(sitebck,warn,m,k1)
                if (warn.eq.1) warnbck=1
                if (z.eq.l-1) then
                        sitetop = 1
                        sitebtm = 2+l*l
                        call correct(sitebtm,warn,m,k1)
                        if (warn.eq.1) warnbtm=1
                elseif (z.eq.1) then
                        sitetop = 2+(l-3)*l*l
                        call correct(sitetop,warn,m,k1)
                        if (warn.eq.1) warntop=1
                        sitebtm = 2+(l-1)*l**2-k1
                        call correct(sitebtm,warn,m,k1)
                        if (warn.eq.1) warnbtm=1
                else
                        sitetop = 2+(l-2-z)*l*l
                        call correct(sitetop,warn,m,k1)
                        if (warn.eq.1) warntop=1
                        sitebtm = 2+(l-z)*l*l
                        call correct(sitebtm,warn,m,k1)
                        if (warn.eq.1) warnbtm=1
                endif
                asum = 0.0
                if (warnlef .ne. 1) then
                        ileft = siteino-sitelef+k
```

```
                        a(ileft,sitelef)=-1.0*const*dia(0,0,z,1)**3.0
                        asum = asum-a(ileft,sitelef)
                else
                        warnlef=0
                endif
                if (warnrit .ne. 1) then
                        iright = siteino-siterit+k
                        a(iright,siterit)=-1.0*const*dia(0,1,z,1)**3.0
                        asum = asum-a(iright,siterit)
                else
                        warnrit=0
                endif
                if (warnfrt .ne. 1) then
                        ifront = siteino-sitefrt+k
                        a(ifront,sitefrt)=-1.0*const*dia(0,0,z,2)**3.0
                        asum = asum-a(ifront,sitefrt)
                else
                        warnfrt=0
                endif
                if (warnbck .ne. 1) then
                        iback = siteino-sitebck+k
                        a(iback,sitebck)=-1.0*const*dia(l-1,0,z,2)**3.0
                        asum = asum-a(iback,sitebck)
                else
                        warnbck=0
                endif
                if (warntop .ne. 1) then
                        itop = siteino-sitetop+k
                        a(itop,sitetop)=-1.0*const*dia(0,0,z+1,3)**3.0
                        asum = asum-a(itop,sitetop)
                else
                        warntop=0
                endif
                if (warnbtm .ne. 1) then
                        ibtm = siteino-sitebtm+k
                        a(ibtm,sitebtm)=-1.0*const*dia(0,0,z,3)**3.0
                        asum = asum-a(ibtm,sitebtm)
                else
                        warnbtm=0
                endif
                a(k,siteino) = asum
300     continue
c
c       Pressure equations formulated for the top node
c
        asum = 0.0
        z=l-1
        do 700 x=0,l-1
        do 700 y=0,l-1
                siteno = l*x+y+2
                call correct(siteno,warn,m,k1)
                if (warn.eq.1) goto 700
                isite = 1-siteno+k
                a(isite,siteno)=-1.0*const*dia(x,y,l,3)**3.0
                asum = asum-a(isite,siteno)
700     continue
c
        a(k,1) = asum
        asum = 0.0
c
c       Pressure equations formulated for the bottom node
c
        z=1
        do 800 x=0,l-1
```

```
      do 800 y=0,l-1
            siteno = l*x+y+2+(l-2)*l*l
            call correct(siteno,warn,m,k1)
            if (warn.eq.1) goto 800
            isite = 2+(l-1)*l*l-k1-siteno+k
            a(isite,siteno)=-1.0*const*dia(x,y,1,3)**3.0
            asum = asum-a(isite,siteno)
800   continue
c
      a(k,2+(l-1)*l*l-k1) = asum
      asum = 0.0
c
c
c     Pressure equations formulated for the nodes inside
c
      do 900 z=l-1,1,-1
      do 900 x=1,l-2
      do 900 y=1,l-2
            siteino = l*x+y+2+(l-1-z)*l*l
            call correct(siteino,warn,m,k1)
            if (warn.eq.1) goto 900
            sitelef = l*x+y+1+(l-1-z)*l*l
            call correct(sitelef,warn,m,k1)
            if (warn.eq.1) warnlef=1
            siterit = l*x+y+3+(l-1-z)*l*l
            call correct(siterit,warn,m,k1)
            if (warn.eq.1) warnrit=1
            sitefrt = (x+1)*l+y+2+(l-1-z)*l*l
            call correct(sitefrt,warn,m,k1)
            if (warn.eq.1) warnfrt=1
            sitebck = (x-1)*l+y+2+(l-1-z)*l*l
            call correct(sitebck,warn,m,k1)
            if (warn.eq.1) warnbck=1
            if (z.eq.l-1) then
                  sitetop = 1
                  sitebtm = l*x+y+2+l*l
                  call correct(sitebtm,warn,m,k1)
                  if (warn.eq.1) warnbtm=1
            elseif (z.eq.1) then
                  sitetop = l*x+y+2+(l-3)*l*l
                  call correct(sitetop,warn,m,k1)
                  if (warn.eq.1) warntop=1
                  sitebtm = 2+(l-1)*l*l-k1
                  call correct(sitebtm,warn,m,k1)
                  if (warn.eq.1) warnbtm=1
            else
                  sitetop = l*x+y+2+(l-2-z)*l*l
                  call correct(sitetop,warn,m,k1)
                  if (warn.eq.1) warntop=1
                  sitebtm = l*x+y+2+(l-z)*l*l
                  call correct(sitebtm,warn,m,k1)
                  if (warn.eq.1) warnbtm=1
            endif
            asum = 0.0
            if (warnlef .ne. 1) then
                  ileft = siteino-sitelef+k
                  a(ileft,sitelef)=-1.0*const*dia(x,y,z,1)**3.0
                  asum = asum-a(ileft,sitelef)
            else
                  warnlef=0
            endif
            if (warnrit .ne. 1) then
                  iright = siteino-siterit+k
                  a(iright,siterit)=-1.0*const*dia(x,y+1,z,1)**3.0
```

```
                        asum = asum-a(iright,siterit)
                else
                        warnrit=0
                endif
                if (warnfrt .ne. 1) then
                        ifront = siteino-sitefrt+k
                        a(ifront,sitefrt)=-1.0*const*dia(x,y,z,2)**3.0
                        asum = asum-a(ifront,sitefrt)
                else
                        warnfrt=0
                endif
                if (warnbck .ne. 1) then
                        iback = siteino-sitebck+k
                        a(iback,sitebck)=-1.0*const*dia(x-1,y,z,2)**3.0
                        asum = asum-a(iback,sitebck)
                else
                        warnbck=0
                endif
                if (warntop .ne. 1) then
                        itop = siteino-sitetop+k
                        a(itop,sitetop)=-1.0*const*dia(x,y,z+1,3)**3.0
                        asum = asum-a(itop,sitetop)
                else
                        warntop=0
                endif
                if (warnbtm .ne. 1) then
                        ibtm = siteino-sitebtm+k
                        a(ibtm,sitebtm)=-1.0*const*dia(x,y,z,3)**3.0
                        asum = asum-a(ibtm,sitebtm)
                else
                        warnbtm=0
                endif
                a(k,siteino) = asum
900     continue
c
c       Solve the nodal pressure matrix using CRAY subroutines
c
c       The matrix A has already been stored in band storage form
c
c       do 1 i=1,(l-1)*l*l+2-k1
c               write(9,*) a(k,i)
c1      continue
c
        call sgbtrf(m1,n,kl,ku,a,lda,ipiv,info)
        print*, info
c
        call sgbtrs(trans,n,kl,ku,nrhs,a,lda,ipiv,b,ldb,info)
c
c       b(i) stores the pressure values for the lattice
c
        stop
        end

        subroutine correct(site,warn,m,k1)
c
c       This program removes nodes which have no bonds attached to it
c       and correct for the singularities
c
        integer site,warn,k1
        integer m(k1)
c
        warn = 1
c
        if (site .ge. 1 .and. site .le. m(1)-1) then
```

```
                site = site
                warn = 0
                return
        elseif (site .gt. m(k1) .and. site .lt. 15002) then
                site = site-k1
                warn = 0
                return
        endif
c
        do 10 i =1,k1-1
                if (site .gt. m(i) .and. site .le. m(i+1)-1) then
                        site = site-i
                        warn = 0
                        return
                endif
10      continue
c
        return
        end
```