Small-Angle X-ray and Neutron Scattering Studies of 3-component Microemulsion and Micellar Solution of Semifluorinated Copolymers

by

Chwen-Yuan Ku
M.S., National Tsing-Hua University (1987)
B.S., National Tsing-Hua University (1985)

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

Doctor of Philosophy in Nuclear Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1996

Author .....
Department of Nuclear Engineering
January 24, 1996

Certified by .....
Sow-Hsin Chen
Professor
Thesis Supervisor

Certified by .....
Xiaolin Zhou
Assistant Professor
Thesis Supervisor

Certified by .....
Anne Mayes
Assistant Professor
Thesis Reader

Accepted by .....
Jeffrey P. Freidberg
Chairman, Departmental Committee on Graduate Students

Small-Angle X-ray and Neutron Scattering Studies of 3-component Microemulsion and micellar solution of Semifluorinated Copolymers

by

Chwen-Yuan Ku

Submitted to the Department of Nuclear Engineering
on February 5, 1996, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Nuclear Engineering

Abstract

Small-Angle Neutron and X-Ray Scattering (SANS & SAXS) are very powerful tools in the studies of microstructure of various materials. In this thesis, SANS and SAXS are used to study the ionic surfactant(AOT)/water/decane microemulsion and an interesting semifluorinated alkane(F_8H_{16}) /perfluorooctane /iso-octane system, respectively. From the SANS data analyses based on the sticky-sphere model of Baxter, we are able to obtain the temperature dependent structural information of the microemulsion system and construct a transformation relation. This relation enables us, for the first time, to explain the experimental phase boundary and percolation line with Baxter's model. SAXS data analyses confirms quantitatively the detailed structural models for the liquid and gel phases of the F_8H_{16} system that we propose based on the qualitative information provided by light scattering and birefringence. From the dimensions extracted from the data analyses, we propose a mechanism for the growth of the gel in the system.

Thesis Supervisor: Sow-Hsin Chen
Title: Professor

Thesis Supervisor: Xiao-Lin Zhou
Title: Assistant Professor

Acknowledgments

I have been very fortunate to be able to come to MIT and to have the opportunities to meet and spend time with many professors and friends here. Without the guidance, support, and help from them, I could not have finished my thesis.

I would like to first thank Professor Sow-Hsin Chen for supervising my graduate studies and research. He has given me a model of a devoted scientist with warm personality and enthusiasm for science to follow in the coming days of my research career. I am also indebted to my co-supervisor, Professor Xiaolin Zhou, for exposing me to the field of reflectometry, and to the rest of my thesis committee, Professor Sidney Yip, Professor Elias Gyftopoulos, and Professor Anne Mayes, for their patience of going through this thesis.

I would also like to thank my collaborators Piero Lo Nostro, Yingchun Liu, J. S. Lin, Dieter Schneider, John Barker, and Bill Orts for assistance and discussions on experiments and theories during these past few years.

There are also many friends, Dan Lee, Lisa Park, Kuang Han Chen, Chen An Chen, Der Hua Peir, Wei Chao, and Pei Ching Chu, who have made my stay at MIT a very enjoyable one.

I am also deeply grateful to my sisters for their constant support from the other side of the earth. Finally, I would like to express my gratitude to my wife, Hui, for all the love she has been giving me during the better and the worse of our life at MIT.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Small-Angle Neutron and X-Ray Scattering

Small-angle scattering of X-rays and neutrons is a widely used diffraction method for studying the structure of matter. This method of elastic scattering is used in various branches of science and technology, including condensed matter physics, molecular biology and biophysics, polymer science, and metallurgy. Many small-angle scattering studies are of value for pure science and practical applications.

It is well known that the most general and informative method for investigating the spatial structure of matter is based on wave-diffraction phenomena. In diffraction experiments a primary beam of radiation influences a studied object, and the scattering pattern is analyzed. In principle, this analysis allows one to obtain information on the structure of a substance with a spatial resolution determined by the wavelength of the radiation.

Diffraction methods are used for studying matter on all scales, from atoms to macro-objects. The use of X rays, neutrons and electron beams, with wavelengths of about $1\mathring{A}$, permits the study of the condensed state of matter, solids and liquids, down to atomic resolution. Determination of the atomic structure of crystals, i.e., the arrangement of atoms in a unit cell, is an important example of this line of investigation. Another line deals with the study of the structure of matter at superatomic level, i.e., with a spatial resolution from $10\mathring{A}$ up to thousands and even several tens

of thousands of angstroms. For this line of investigation the basic tool is small-angle scattering, mainly of X-rays and neutrons.

In small-angle scattering research the wavelength used is, as a rule on the order of several angstroms. As interatomic distances are of same order of magnitude, the diffraction pattern, corresponding to the superatomic structure, lies in the small-angle region, and this is how the method received its name. The most important feature of the small-angle scattering method is its potential for analyzing the inner structure of disordered systems, and frequently the application of this method is a unique way to obtain direct structural information on systems with random arrangement of density inhomogeneities of colloidal size, namely, $10^1 - 10^4 \mathring{A}$.

Small-angle scattering studies date from the classical works of A. Guinier[1]. Later, G. Porod[2], O. Kratky[3], V. Luzzati[4], P. Schmidt[5], and others developed the theoretical and experimental fundamentals of the method[6, 7]. The principles of designing small-angle scattering facilities were developed, and the potential for applying the method for determining general structural characteristics of various types of highly dispersed systems was shown.

New progress in refining the method of small-angle scattering began in 1970s and is continuing today. This state of work is characterized by new opportunities both for experimentation (powerful neutron beams, synchrotron X-ray sources, position-sensitive detectors) and for structural interpretation(contrast variation, isomorphous replacement, direct methods analysis of characteristic functions). Certainly, the use of various computers is of great significance in advancing these two aspects of small-angle scattering applications. The result has been a gradual expansion of the range of studied objects and an increase in the spatial resolution in both directions. Thus, for example, the mechanisms of phase separation and the sizes and degrees of dispersion of dispersed structures of alloys, powders, and glasses, the specific features of the configuration polymer chains in different aggregate states, and the geometrical and weight parameters(and, sometimes, three-dimensional structure) of biological macromolecules and their complexes in solution can all be the subjected to analysis.

Currently, the small-angle scattering technique, with its well-developed exper-

imental and theoretical procedures add wide range of studied objects, is a self-contained branch of the structural analysis of matter[8].

1.2 Resolution Function

Traditionally, most small-angle X-ray scattering experiments have been performed in a long slit geometry, using the "Kratky" type of camera[7]. Numerous studies have been made in order to perform the desmearing analysis which is necessary in connection with this technique[9, 10, 11]. Small-angle neutron scattering experiments are, on the other hand, always performed in a pinhole geometry, and area-sensitive detectors are widely used. The smearing effects for pinhole geometry are of course much smaller than for long-slit geometry as long as the wavelength spread of the incident beam is small. Therefore the experimental data measured in the pinhole geometry are usually analyzed without taking any smearing effects into account. However, the limited resolution of the instrument will lead to some smearing which might have an important influence on the measured scattering data.

Some numerical studies have already been made in order to investigate the smearing effects[12, 13]. However, it is found there is a great need for an analytical treatment of the resolution effects. This would allow these effects to be incorporated in a relatively simply way in the analysis of the experimental data. This is further motivated by the fact that at present there is a tendency also to use pinhole geometry and area-sensitive detectors in new X-ray sources such as the rotating-anode sources and synchrotron X-ray facilities.

The resolution effect is described by the resolution function $R(\vec{q}, \langle\vec{q}\rangle)$, where $\langle\vec{q}\rangle$ is the average scattering vector corresponding to the setting of the instrument ($\langle\rangle$ means average value of a parameter). The length of the scattering vector is $\langle q\rangle = 2\langle k\rangle \sin\langle\theta\rangle$, where $2\langle theta\rangle$ is the scattering angle, $\langle k\rangle$ is the length of the wavevector, $\langle k\rangle = 2\pi/\langle\lambda\rangle$, where $\langle\lambda\rangle$ is the wavelength of the neutrons or X-rays. When the instrument is set to detect radiation with scattering vector $\langle\vec{q}\rangle$, radiation with scattering vectors $\vec{q}$ in a range around $\langle\vec{q}\rangle$ also contributes to the scattering due to the finite collimation, the

spread in the wavelength of the incident beam and the finite spatial resolution of the detector. The resolution function $R(\vec{q}, \langle\vec{q}\rangle)$ describes the distribution of the radiation with scattering vector $\vec{q}$ contributing to the scattering for setting $\langle\vec{q}\rangle$. According to this the measured intensity at $\langle\vec{q}\rangle$ is proportional to

$$I(\langle\vec{q}\rangle) = \int R(\vec{q}, \langle\vec{q}\rangle)\frac{d\sigma(\vec{q})}{d\Omega}d\vec{q} \tag{1.1}$$

Where $d\sigma(\vec{q})/d\Omega$ is the scattering cross section.

Previous investigations of the properties of $R(\vec{q}, \langle\vec{q}\rangle)$ reported in the literature have mainly been connected with the optimization of the dimensions of small-angle scattering instruments in order to give the highest flux for a given resolution[14, 57, 16, 17, 18, 19]. In other publications the instrumental resolution has been treated more in its own right[20, 12, 21, 13, 11, 22]. Investigations of the width of the resolution function have usually been done analytically by calculating variances and the results have been supported by computer simulation. General equations for the smearing of an ideal scattering curve due to resolution effects have been given by Moore[23], Miller et al.[20], Ramakrishnan[12] and Schmidt[11]. These works led to equations which express the smearing in terms of integrals involving weighting functions describing the transmission of the collimating system and the wavelength distribution. Expressions for the weighting function have to be determined either experimentally or by calculations[24, 25, 12], and practical applications involve numerical calculation of several integrals. In this chapter, the different contributions to the resolution function are all approximated by Gaussian functions. This allows the integrals connected with the combination of the different contributions to the resolution function to be performed analytically. One obtains the resolution function expressed as a Gaussian function on a two-dimensional $\vec{q}$ space. The azimuthal-integrated resolution function to be applied for radial-averaged scattering profiles can also be calculated analytically[26]. Hence, smearing of an ideal scattering curve involves in this case only one numerical integration.

The Gaussian function describing the various contributions to the resolution func-

tion are determined when the parameters describing the width and the normalization condition are chosen. It has been decided to define the Gaussians to have the same full-width-at half-maximum value as the distributions they approximate and to give unity when integrated over the two-dimensional $\vec{q}$ space[27]. However, other criteria may be applied. For example, the variances of the Gaussians could have been chosen to have the same value as the distributions they approximate[17].

The analytical treatment of the resolution function is begun with deriving separately the resolution functions resulting from (1) wavelength spread, (2) collimation effects, and (3) detector resolution. In each of these derivations the two other contributions not under consideration are assumed to be negligible. Then the combined resolution due to all three contributions are calculated, assuming that they are independent. Finally, the resolution function for radial-averaged data are derived.

1.2.1 Resolution Function due to Wavelength Spread

Assume that the wavelength distribution can be described by a Gaussian function centered on wavelength $\langle\lambda\rangle$ having a full-width-at half-maximum value (FWHM) $\Delta\lambda$:

$$F(\lambda, \langle\lambda\rangle) = [(2\pi)^{1/2}\sigma_\lambda]^{-1} \exp[-(\lambda - \langle\lambda\rangle)^2/2\sigma_\lambda^2] \tag{1.2}$$

where σ_λ is related to the FWHM value $\Delta\lambda$ of the distribution by

$$\sigma_\lambda = \Delta\lambda/[2(2\ln 2)^{1/2}] \tag{1.3}$$

A first-order Taylor expansion of expression for the scattering vector with respect to $\lambda - \langle\lambda\rangle$ gives

$$q - \langle q\rangle = -(\langle q\rangle/\langle\lambda\rangle)(\lambda - \langle\lambda\rangle) \tag{1.4}$$

and multiplication of nominator and denominator of the exponent in Eq.1.2 by $(\langle q\rangle/\langle\lambda\rangle)^2$ gives

$$R_W(q, \langle q \rangle) = [(2\pi)^{1/2}\sigma_W]^{-1} \exp[-(q - \langle q \rangle)^2/2\sigma_W^2] \tag{1.5}$$

where

$$\sigma_W = \sigma_\lambda \frac{\langle q \rangle}{\langle \lambda \rangle} = \langle q \rangle \frac{\Delta\lambda}{\langle \lambda \rangle} \frac{1}{2(2\ln 2)^{1/2}} \tag{1.6}$$

Note that the width σ_W of R_W is proportional to $\langle q \rangle$. The prefactor of the exponential function in Eq.1.5 has been chosen to give $R_W(q, \langle q \rangle)$ the proper normalization:

$$\int_0^\infty R_W(q, \langle q \rangle) dq = 1 \tag{1.7}$$

The measured intensity is proportional to

$$I(\langle \vec{q} \rangle) = \int_0^\infty R_W(q, \langle q \rangle) \frac{d\sigma(\vec{q})}{d\Omega} dq \tag{1.8}$$

Where $\vec{q}$ is parallel to $\langle \vec{q} \rangle$. In Eq.1.8 the integration element is one-dimensional, in contrast to the integration element in Eq.1.1 which is two-dimensional.

In a small-angle X-ray scattering (SAXS) experiment the incident beam is monochromatized by Bragg scattering from a crystal. For a conventional X-ray source the crystal is set to scatter one of the characteristic lines of the source and the wavelength spread is a result of the combined effects of the natural line width, the mosaic spread or the Darwin width of the monochromator crystal and the collimation before and after the crystal. The spectrum of the radiation from a synchrotron is continuous and the wavelength spread is only determined by the mosaic spread or Darwin width of the monochromator and the collimation. However, for both the conventional an synchrotron X-ray source the wavelength spread $\Delta\lambda/\langle\lambda\rangle$ is typically smaller than 10^{-3} and it can therefore be neglected in most cases.

For small-angle neutron scattering (SANS) experiments the relatively low flux available makes the use of monochromators with a relatively large wavelength spread necessary. This can be either a segmented monochromator crystal[13] or mechanical velocity selector[28, 29].

1.2.2 Resolution Function due to Finite Collimation

When the range is chosen sufficiently small in a small range around $\langle\vec{q}\rangle = 0$, changes in the shape and the width of the resolution can be neglected. Hence, it is only dependent on the difference $|\vec{q} - \langle\vec{q}\rangle|$, i.e., we can specifically write $R_c(\vec{q}, \langle\vec{q}\rangle) = R_c^0(|\vec{q} - \langle\vec{q}\rangle|)$. The unscattered beam can be described by the cross section $d\sigma(\vec{q})/d\Omega = \delta(\vec{q})$, where δ is a delta function, and Eq.1.1 gives the result that the measured intensity is proportional to

$$I(\langle\vec{q}\rangle) = R_c^0(|\langle\vec{q}\rangle|) \tag{1.9}$$

This shows that the collimation-related resolution close to $\langle\vec{q}\rangle = 0$ can be determined by measuring the profile of the direct beam. This fact is used to derive the resolution function for small values of $\langle q\rangle$ by calculating the width and shape of the direct beam. A schematic drawing of the small-angle scattering setup is shown in Fig.1-1. The first aperture is denoted as the source aperture and the second the defining aperture. In an experiment the sample is situated just after the defining aperture. Here it is assumed that the sizes of the apertures are small compared with their separation L and the distance l between the defining aperture and the detector plane. First-order approximations for the opening angles can then be applied in the calculation of the collimation effects. In the general situation it is assumed that the apertures are circular and that they are uniformly illuminated.

For simplicity, it is started by considering a special situation with negligible extension of the apertures in the direction perpendicular to the plane of the paper in Fig.1-1. From a point on the detector half the angular extent of the apertures is $a_1 = r_1/(l + L)$ and $a_2 = r_2/l$ for the source and defining aperture, respectively (see Fig.1-2(a)). The intensity at a point on the detector is determined by the angular overlap of the two apertures. The intensity is expressed as a function of the angle β, shown in Fig.1-1, which corresponds to a scattering angle. By this choice it is straightforward to convert from angle to scattering vector.

At the point on the detector corresponding to the angle β, the angle ν between

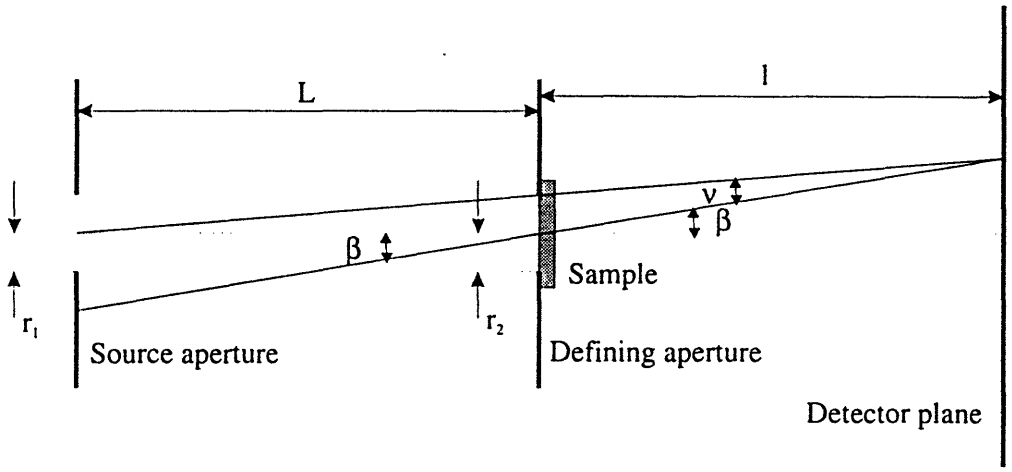

Figure 1-1: A schematic representation of the setup. The plane of the paper corresponds to the scattering plane. The radius of the source and the defining apertures are r_1 and r_2, respectively. The distance between the apertures is L and the distance between the defining aperture and the detector plane is l. Lines have been drawn from a point on the detector through the center of the two apertures to define the various angles referred to in the text. The figure demonstrates the relation $\nu = \beta L/(L+1)$ of ν to the scattering angle β.

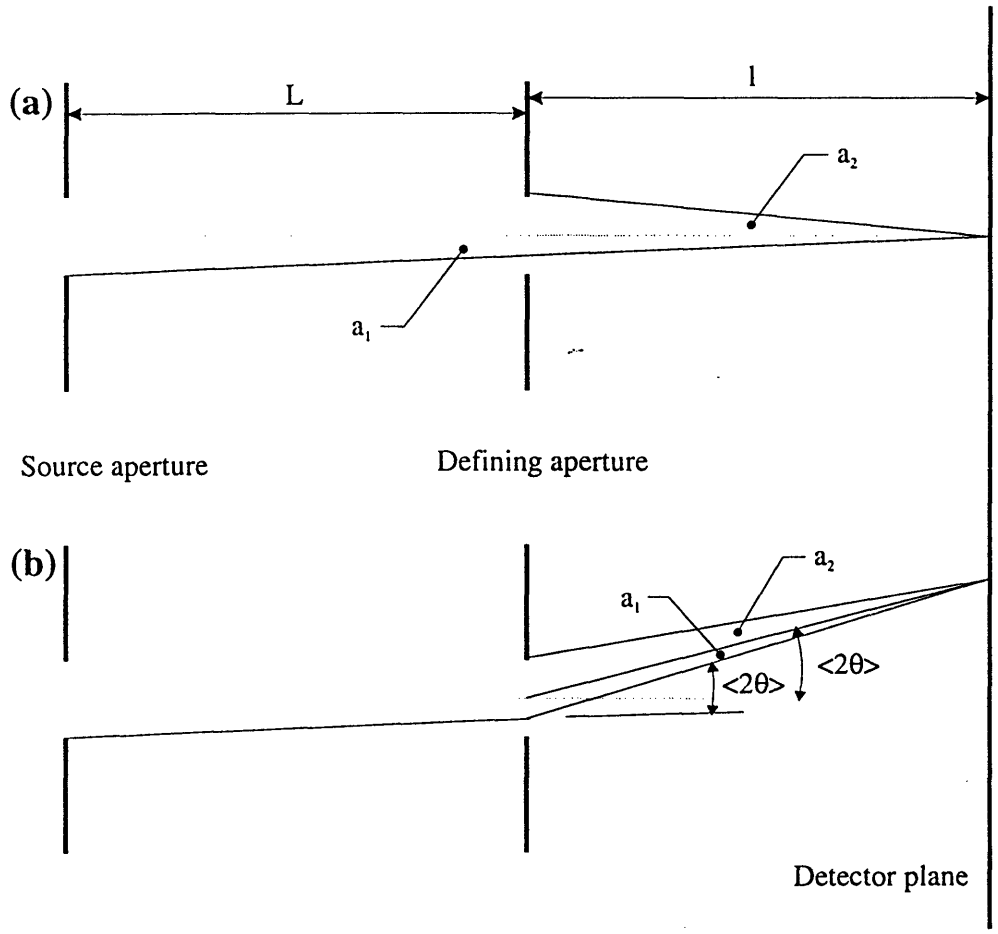

Figure 1-2: (a) The angular extent of the source and defining apertures, a_1 and a_2, respectively, seen from a point on the detector. (b) The effective in-plane angular extent of the apertures for a beam scattered through the angle $\langle 2\theta \rangle$.

the centers of the source and the defining aperture is $\nu = \beta L/(L+l)$. This equation gives the transformation on between the angular separation ν and the angle β. For $a_1 = a_2$ the overlap as a function of β is a triangular function and for $a_1 \neq a_2$ the function has a trapezoidal shape. The FWHM value $\Delta\beta$ is determined by the angle where the angular overlap, and thereby the intensity, is reduced to half the maximum value:

$$\begin{aligned} \Delta\beta &= 2r_1/L && \text{For } a_1 \geq a_2 \\ &= 2r_2(1/l + 1/L) && \text{For } a_1 < a_2 \end{aligned} \tag{1.10}$$

For circular apertures the overlap is slightly more complicated to calculate. Fig.1-3 shows the situation for $a_1 > a_2$ at an angle corresponding to the FWHM value for apertures with negligible extent perpendicular to the scattering plane. A correction to the FWHM values will be determined in Eq.1.10 for circular apertures. The angle γ defined in Fig.1-3 can be approximated by $\gamma \approx \tan\gamma = a_2/a_1$. A better value for $\Delta\beta$ is obtained using $\frac{1}{2}a_1(1+\cos\gamma)$ instead of a_1 in Eq.1.10. There is $\frac{1}{2}a_1(1+\cos\gamma) \approx a_1 - \frac{1}{4}a_2^2/a_1$. Insertion of the expressions for a_1 and a_2 and multiplication by $(L+l)/L$ to convert to a β angle gives the FWHM values

$$\begin{aligned} \Delta\beta &= \tfrac{2r_1}{L} - \tfrac{lr_2^2}{2r_1}\tfrac{(l+L)^2}{l^2}\tfrac{1}{L} && \text{For } a_1 \geq a_2 \\ &= 2r_2\left(\tfrac{1}{l} + \tfrac{1}{L}\right) - \tfrac{1}{2}\tfrac{r_1^2}{r_2}\tfrac{l}{L}\tfrac{1}{l+L} && \text{For } a_1 < a_2 \end{aligned} \tag{1.11}$$

These equations are valid for small values of $\langle q \rangle$. The distribution is approximated by a Gaussian function with the same width. From a purely qualitative judgment it is estimated that this approximation is acceptable when a_1 and a_2 deviate by less a factor of two. For $a_1 \gg a_2$ (or $a_2 \gg a_1$) the true distribution approaches a "box function" which is poorly described by a Gaussian function.

Next the FWHM values of the distribution function for an arbitrary value of $\langle \vec{q} \rangle \neq 0$ is derived. It is assumed that we have a sample with an ideal Bragg reflection at the reciprocal-lattice vector $\vec{q_0}$. This can be described by the cross section $d\sigma(\vec{q})/d\Omega = \delta(\vec{q} - \vec{q_0})$. A sufficiently small range around $\langle \vec{q} \rangle = \vec{q_0}$ is considered to allow changes in the resolution function to be neglected. The resolution function is then

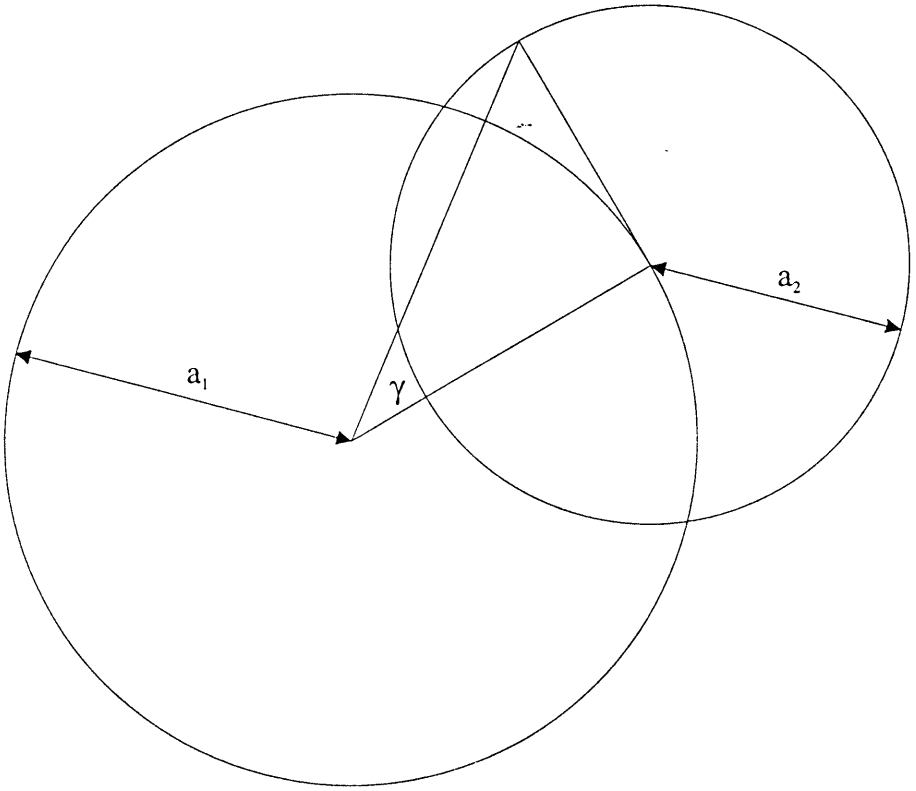

Figure 1-3: The figure defines the angle γ and $\gamma \approx \tan\gamma = a_2/a_1$.

only dependent on the difference $\vec{q} - \langle\vec{q}\rangle$. The measured intensity can be calculated by Eq.1.1:

$$I(\langle\vec{q}\rangle) = R_c^{q_0}(\vec{q_0} - \langle\vec{q}\rangle|) \tag{1.12}$$

Accordingly, the resolution function around $\langle\vec{q}\rangle = \vec{q_0}$ can be determined by measuring the actual intensity distribution of the Bragg reflection. This will be used in our derivation and the resolution function will be calculated by calculating the width of the corresponding Bragg reflection. In the derivation it is implicitly assumed the scattering vectors to be two-dimensional vectors, thereby relating points on the two-dimensional detector to the two components of the scattering vector. For the scattering situation in reciprocal space, the third dimension is in the direction perpendicular to the Ewald sphere. For scattering from disordered or polycrystalline materials this has no consequences as the third dimension is effectively integrated out of the problem. However, for crystalline materials it is also necessary to consider whether the Bragg condition is fulfilled in the direction perpendicular to the Ewald sphere and in order to measure the integrated intensity it is necessary to perform a sample rotation scan along an axis normal to the scattering plane. Therefore, when actually measuring the resolution function using a Bragg reflection, this rotation has to be performed. In the derivation of the resolution function this is taken into account by assuming the beam to be scattered through the same $2\theta_0$ in the plane independent of the direction of the incident beam.

Consider the neutrons or X-rays which are scattered through the angle $2\theta_0 = 2\arcsin[q_0/(2\langle k\rangle)] = \langle 2\theta\rangle$. The shape of the Bragg reflection can be calculated as the angular overlap between the source and the defining aperture, taking the scattering into account. Again we start by considering a situation for which the apertures have negligible extent in the direction perpendicular to the scattering plane(Fig.1-2). Half the angular extent of the source aperture is effectively $a_1 = r_1/(L + l/\cos^2\langle 2\theta\rangle)$ and for the defining aperture $a_2 = r_2\cos^2\langle 2\theta\rangle/l$. The derivation follows the same lines as in the first part of this section. The results for the FWHM values of the angles in

scattering plane are

$$\begin{aligned}\Delta\beta_1 &= \frac{2r_1}{L} - \frac{lr_2^2}{2r_1}\frac{\cos^4\langle 2\theta\rangle}{l^2 L}\left(L + \frac{l}{\cos^2\langle 2\theta\rangle}\right)^2 && \text{For } a_1 \geq a_2 \\ &= 2r_2\left(\frac{\cos^2\langle 2\theta\rangle}{l} + \frac{1}{L}\right) - \frac{1}{2}\frac{r_1^2}{r_2}\frac{l}{L}\frac{1}{\cos^2\langle 2\theta\rangle\left(L + \frac{l}{\cos^2\langle 2\theta\rangle}\right)} && \text{For } a_1 < a_2\end{aligned} \tag{1.13}$$

The corresponding results for the distribution in the direction perpendicular to the scattering plane can be calculated by considering a situation for which the apertures have negligible extent in the scattering plane. The angular extents of the slits are $a_1 = r_1/(L + l/\cos\langle 2\theta\rangle)$ and $a_2 = r_2\cos\langle 2\theta\rangle/l$, and this gives the FWHM values

$$\begin{aligned}\Delta\beta_2 &= \frac{2r_1}{L} - \frac{lr_2^2}{2r_1}\frac{\cos^2\langle 2\theta\rangle}{l^2 L}\left(L + \frac{l}{\cos\langle 2\theta\rangle}\right)^2 && \text{For } a_1 \geq a_2 \\ &= 2r_2\left(\frac{\cos\langle 2\theta\rangle}{l} + \frac{1}{L}\right) - \frac{1}{2}\frac{r_1^2}{r_2}\frac{l}{L}\frac{1}{\cos\langle 2\theta\rangle\left(L + \frac{l}{\cos\langle 2\theta\rangle}\right)} && \text{For } a_1 < a_2\end{aligned} \tag{1.14}$$

As mentioned previously, the distribution function describing the collimation effects is taken to be Gaussian and the total distribution is the product of the in-plane F_1 and out-of-pane F_2 distribution functions:

$$\begin{aligned}F_1(2\theta, \langle 2\theta\rangle) &= [(2\pi)^{1/2}\sigma_\theta]^{-1}\exp[-\tfrac{1}{2}(2\theta - \langle 2\theta\rangle)^2/\sigma_\theta^2] \\ F_2(\delta, 0) &= [(2\pi)^{1/2}\sigma_\delta]^{-1}\exp[-\tfrac{1}{2}\delta^2/\sigma_\delta^2]\end{aligned} \tag{1.15}$$

where $(2\theta, \langle 2\theta\rangle)$ and δ are the deviations in scattering plane and perpendicular to the scattering plane, respectively, and

$$\begin{aligned}\sigma_\theta &= \Delta\beta_1/2(2\ln 2)^{1/2} \\ \sigma_\delta &= \Delta\beta_2/2(2\ln 2)^{1/2}\end{aligned} \tag{1.16}$$

The first-order Taylor expansion of the scattering vector $\vec{q}$ around $\langle\vec{q}\rangle$ can now be made:

$$\vec{q} - \langle\vec{q}\rangle = \begin{pmatrix} q_1 - \langle q\rangle \\ q_2 \end{pmatrix} \approx \begin{pmatrix} 2\langle k\rangle\cos\langle\theta\rangle(\theta - \langle\theta\rangle) \\ \langle k\rangle\delta \end{pmatrix} \tag{1.17}$$

Where q_1 is the component of $\vec{q}$ parallel to $\langle\vec{q}\rangle$ and q_2 is the component perpen-

dicular to $\langle\vec{q}\rangle$.

Expressing the distribution function Eq.1.15 in terms of this gives

$$\begin{aligned} F_1(q_1,\langle q\rangle) &= [(2\pi)^{1/2}\sigma_{C1}]^{-1}\exp[-\tfrac{1}{2}(q_1 - langleq\rangle)^2/\sigma_{C1}^2] \\ F_2(q_2,0) &= [(2\pi)^{1/2}\sigma_{C2}]^{-1}\exp[-\tfrac{1}{2}q_2^2/\sigma_{C2}^2] \end{aligned} \tag{1.18}$$

where

$$\begin{aligned} \sigma_{C1} &= \langle k\rangle\cos\langle\theta\rangle\Delta\beta_1/2(2\ln 2)^{1/2} \\ \sigma_{C2} &= \langle k\rangle\Delta\beta_2/2(2\ln 2)^{1/2} \end{aligned} \tag{1.19}$$

The resolution function is the product of F_1 and F_2:

$$R_C(\vec{q},\langle\vec{q}\rangle) = [2\pi\sigma_{C1}\sigma_{C2}]^{-1}\exp\left[-\frac{1}{2}\left(\frac{(q_1-\langle q\rangle)^2}{\sigma_{C1}^2}+\frac{q_2^2}{\sigma_{C2}^2}\right)\right] \tag{1.20}$$

where the prefactor of the exponential function has been chosen to give the proper normalization when the expression is integrated over the two-dimensional $\vec{q}$ space. When the wavelength spread is small and the collimation effects are dominant, Eq.1.20 for the resolution function can be applied. This is the case for X-rays, as mentioned previously.

1.2.3 Resolution Function due to Detector Resolution

Consider a two-dimensional area-sensitive detector. There are usually three contributions to the spatial resolution: the division of the detector into pixels, the method of detection and the method of position determination. It is assumed that the latter two dominate and that the spatial resolution can be described by a Gaussian function of width Δ. First, the width of the resolution function in the direction parallel to $\langle\vec{q}\rangle$ is determined. Let $\langle s\rangle$ denote the distance from the beam center at the detector to the point on the detector with scattering angle $\langle 2\theta\rangle$. Then $\langle\theta\rangle = \frac{1}{2}\arctan(\langle s\rangle/l)$ and a first-order Taylor expansion of the scattering vector with respect to s gives

$$q_1 - \langle q\rangle = \langle k\rangle\cos\langle\theta\rangle\cos^2\langle 2\theta\rangle(1/l)(s-\langle s\rangle) \tag{1.21}$$

The Gaussian distribution of $(s - \langle s \rangle)$ with the FWHM value Δ gives a Gaussian distribution of $q_1 - \langle q \rangle$ with the width

$$\sigma_{D1} = \langle k \rangle \cos\langle \theta \rangle \cos^2\langle 2\theta \rangle \Delta [l2(2\ln 2)^{1/2}]^{-1} \tag{1.22}$$

Similarly it can be shown that in the direction perpendicular to $\langle \vec{q} \rangle$ the distribution of the projection q_2 of $\vec{q}$ has the width

$$\sigma_{D2} = \langle k \rangle \cos\langle 2\theta \rangle \Delta [l2(2\ln 2)^{1/2}]^{-1} \tag{1.23}$$

Hence the resolution function due to the detector resolution is

$$R_D(\vec{q}, \langle \vec{q} \rangle) = [2\pi\sigma_{D1}\sigma_{D2}]^{-1} \exp\left[-\frac{1}{2}\left(\frac{(q_1 - \langle q \rangle)^2}{\sigma_{D1}^2} + \frac{q_2^2}{\sigma_{D2}^2}\right)\right] \tag{1.24}$$

1.2.4 Combined Resolution Function

In this section the case for which both the wavelength spread and the effects of finite collimation are of significance is treated. Performing a first-order Taylor expansion of $\vec{q}$ as a multi-parameter function around $\langle \vec{q} \rangle$, it can be obtained

$$\begin{aligned} \vec{q} - \langle \vec{q} \rangle &= \begin{pmatrix} q_1 - \langle q \rangle \\ q_2 \end{pmatrix} \\ &= \begin{pmatrix} -\frac{\langle q \rangle}{\langle \lambda \rangle}(\lambda - \langle \lambda \rangle) + 2\langle k \rangle \cos\langle \theta \rangle (\theta - \langle \theta \rangle) \\ \langle k \rangle \delta \end{pmatrix} \end{aligned} \tag{1.25}$$

The in-plane component $q_1 - \langle q \rangle$ is the sum of two term which both have Gaussian distributions (Eq.1.2 and 1.15). Therefore the distribution of $q_1 - \langle q \rangle$ is also Gaussian with the width

$$\sigma^2 = \sigma_W^2 + \sigma_{C1}^2$$

$$= \left[\langle q\rangle \frac{\Delta\lambda}{\langle\lambda\rangle} \frac{1}{2(2\ln 2)^{1/2}}\right]^2 + \left[\frac{\langle k\rangle \cos\langle\theta\rangle \Delta\beta_1}{2(2\ln 2)^{1/2}}\right]^2 \tag{1.26}$$

where σ_W and σ_{Cl} are the widths of the resolution functions due to wavelength spread and finite collimation given in Eq.1.6 and 1.19, respectively. To the order of the calculation the out-of-plane component q_2 does not depend on λ and the distribution is the same as already derived in Eq.1.18. The combined resolution function is[27]

$$R(\vec{q}, \langle\vec{q}\rangle) = (2\pi\sigma\sigma_{C2})^{-1} \exp\left[-\frac{1}{2}\left(\frac{(q_1 - \langle q\rangle)^2}{\sigma^2} + \frac{q_2^2}{\sigma_{C2}^2}\right)\right] \tag{1.27}$$

which has the proper normalization when integrated over the two-dimensional $\vec{q}$ space. Note that the wavelength spread gives rise to a broadening of the resolution function in the direction of $\langle\vec{q}\rangle$. This broadening increases for increasing values of $\langle q\rangle$. To the order of the calculations there is no influence of the wavelength spread in the direction normal to $\langle\vec{q}\rangle$. This result, which expresses the difference of the widths of the resolution function along $\langle\vec{q}\rangle$ and perpendicular to $\langle\vec{q}\rangle$, demonstrates the necessity of making a distinction between these two directions.

Finally, the effects of detector resolution is included. The total resolution function is calculated by convoluting the two contributions described by Eq.1.24 and 1.27. For an Gaussian function the result of the convolution is a Gaussian function with the width squared given by the sum of the widths squared of the two contributions. Hence, the total resolution function is given by Eq.1.27 with σ^2 replaced by $\sigma^2 + \sigma_{D1}^2$ and σ_{C2}^2 replaced by $\sigma_{C2}^2 + \sigma_{D2}^2$.

1.2.5 Resolution Function for Radially-Averaged Data

In many applications of the small-angle scattering technique the scattering pattern is circularly symmetric. Therefore, the two-dimensional integration involved in the convolution of the cross section and the resolution function can be replaced by a one-dimensional integration in analogy with Eq.1.8. This is accomplished by introducing a resolution function integrated along circles where q is constant.

In order to get through the algebra we need to make the assumption that the

resolution function is symmetric ($\sigma_{C1} = \sigma_{C2} = \sigma$) in a range around $\langle\vec{q}\rangle$. This is fulfilled in many cases since the contribution from the wavelength spread vanishes for $\langle q\rangle \to 0$. Specifically, for $\langle q\rangle < q_0$ it is assumed that the resolution function is symmetric and $q_0 \gg \sigma_{Cl}$. Using $q_1 = q\cos\varphi$ where φ is the angle between $\langle\vec{q}\rangle$ and $\vec{q}$, we obtain

$$R(\vec{q}, \langle\vec{q}\rangle) = (2\pi\sigma^2)^{-1} \exp[-\frac{1}{2}(q^2 + \langle q\rangle^2 - 2q\langle q\rangle\cos\varphi)/\sigma^2] \tag{1.28}$$

The radial-averaged resolution function is defined to give the measured intensity proportional to

$$I(\langle q\rangle) = \int_0^\infty R_{av}(q, \langle q\rangle)\frac{d\sigma(q)}{d\Omega}dq \tag{1.29}$$

where

$$\begin{aligned} R_{av}(q, \langle q\rangle) &= \int_{-\pi}^{\pi} R(q, \langle q\rangle) q d\varphi \\ &= (q/\sigma^2)\exp[-\frac{1}{2}(q^2/\sigma^2 + \langle q\rangle^2/\sigma^2)] I_0(q\langle q\rangle/\sigma^2) \end{aligned} \tag{1.30}$$

and I_0 is the modified Bessel function of first kind and zeroth order. This result was first derived by Freltoft, Kjems & Sinha[26]. For $\langle q\rangle$ approaching q_0 the large-argument expression of the Bessel function $I_0(x) = e^x/\sqrt{2\pi x}$ can be used together with $q \approx \langle q\rangle$:

$$R_{av}(q, \langle q\rangle) \approx [(2\pi)^{1/2}\sigma]^{-1} \exp[-\frac{1}{2}(q - \langle q\rangle)^2/\sigma^2)] \tag{1.31}$$

for $\langle q\rangle \to q_0$, $\langle q\rangle < q_0$.

For $\langle q\rangle > q_0$ the resolution function is elongated along $\langle\vec{q}\rangle$ and the integration in Eq.1.30 is not easily performed. It will be shown that one can obtain Eq.1.31 for $\langle q\rangle > q_0$ by changing the integration variables in Eq.1.30. The resolution function is only different from zero in a small range around $\langle q\rangle$ and in this range $q \approx q_1$, where q_1 is the component of $\vec{q}$ along $\langle\vec{q}\rangle$ and $qd\varphi \approx dq_2$, where q_2 is the component of $\vec{q}$

perpendicular to $\langle\vec{q}\rangle$. Therefore, the integration in Eq.1.30 of $qd\varphi$ can be replaced by an integration of dq_2. Another way of expressing the approximation $qd\varphi \approx dq_2$ is that for large values of $\langle q\rangle$ the circles q = constant can, in the range where the resolution function is different from zero, be replace by straight lines in the direction perpendicular to $\langle\vec{q}\rangle$. Using Eq.1.27 for the resolution function and performing an integration of q_2 we can obtain Eq.1.31 as mentioned previously.

In applications it is important that Eq.1.30 is used for small values of q and not Eq.1.31. The two functions deviate systematically for small q values, and the center of mass of Eq.1.30 is at a q value larger than $\langle q\rangle$.

1.2.6 Examples

Fig.1-4 shows the radially-averaged resolution function for a typical situation where $\langle\lambda\rangle = 5.0\AA$, $\Delta\lambda/\langle\lambda\rangle = 0.2$ and $a_1 = a_2 = 0.01$. One gets $\sigma_W = \langle q\rangle\sigma'$, $\sigma' = 0.085$ (Eq.1.6), $\sigma_{C1} \approx \sigma_{C2} \approx 0.008\AA^{-1}$ (Eqs.1.13,1.14,1.19),and the resolution function is symmetric when $\langle q\rangle < q_0 < \sigma_{C1}/\sigma' \approx 0.10\AA^{-1}$. The relation $q_0 \gg \sigma_{C1}$ can easily be fulfilled and it is meaningful to calculate the radial-averaged resolution function. From these considerations it can also be concluded that when the wavelength spread is negligible, the radially-averaged resolution function is given by Eq.1.30 with $\sigma = \sigma_{C1}$.

The 3-component microemulsion system consisting of ionic surfactant AOT/water/decane tends to form a water-in-oil droplet microemulsion near room temperature. AOT is an ionic surfactant with a sodium sulfate hydrophilic head and two hydrophobic hydrocarbon tails. At higher temperature, these droplets percolate and form bicontinuous microemulsion[30]. Fig.1-5 shows a SANS scattering intensity curve of this bicontinuous microemulsion and the data analysis using

$$I_m(\langle q\rangle) = \text{Background} + \langle\eta^2\rangle \int_{\langle q\rangle-3\sigma}^{\langle q\rangle+3\sigma} R_{av}(q,\langle q\rangle)S(q)dq \tag{1.32}$$

where I_m is the measured intensity and

$$S(q) = (1-A_0)\pi^{2/3}c^3e^{-q^2c^2/4} + A_0\frac{8\pi/\xi}{a^2-2b^2q^2+q^4} \tag{1.33}$$

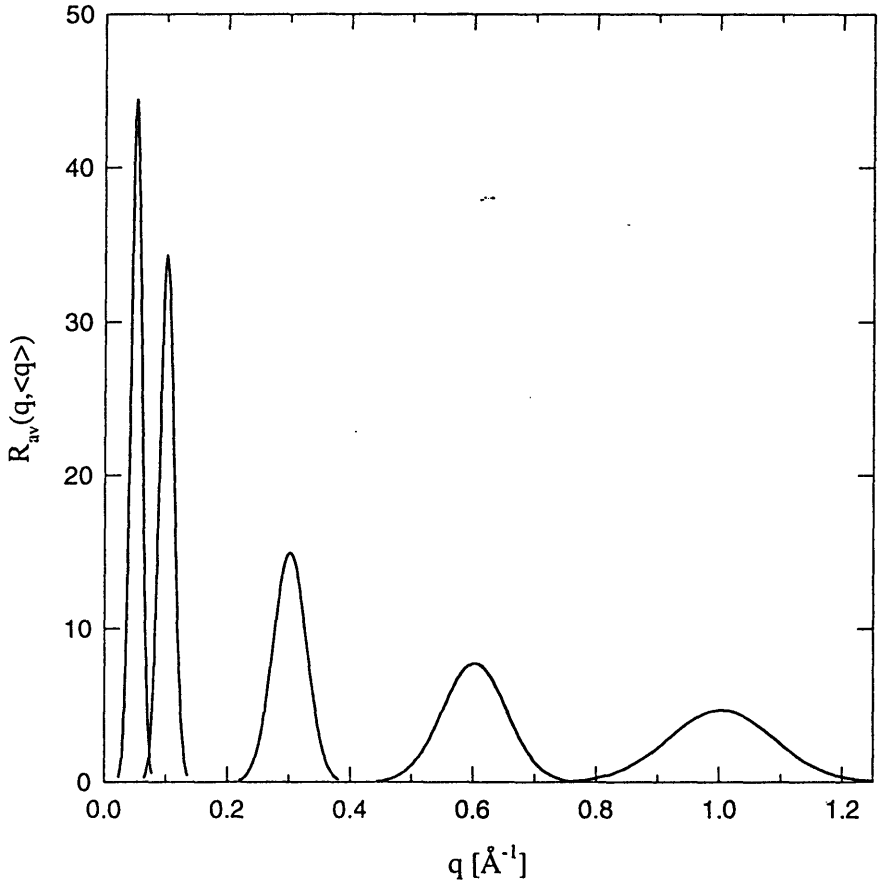

Figure 1-4: The azimuthal-integrated resolution function for $\langle\lambda\rangle = 5.0\mathring{A}$, $\Delta\lambda/\langle\lambda\rangle = 0.2$, $r_1 = 2.0$, $r_2 = 1.0$, $L = l = 200$ cm. The function is plotted for $\langle q\rangle = 0.05$, 0.1, 0.3, 0.6, and $1.0\,\mathring{A}^{-1}$, respectively.

with $a = k^2 + 1/\xi^2$, $b = k^2 - 1/\xi^2$. The SANS experiment was performed at NIST with $\langle\lambda\rangle = 5\,\mathring{A}$, $\Delta\lambda/\langle\lambda\rangle = 0.147$, $r_1 = 0.6\,cm$, $r_2 = 1.9\,cm$, $L = 1,622\,cm$, $l = 310\,cm$, and $\Delta = 1\,cm$. The variables are displayed in the figure.

In Fig.1-5, the dashed line is the calculation with the same variables used to obtain the solid line without the resolution function correction. The smearing by the resolution correction is clearly shown. From Figs.1-5 and 1-4, it is evident that the resolution correction is necessary for scattering curves with sharp features, especially at larger q.

A FORTRAN computer program calculates the resolution function with given parameters is listed in Appendix A.

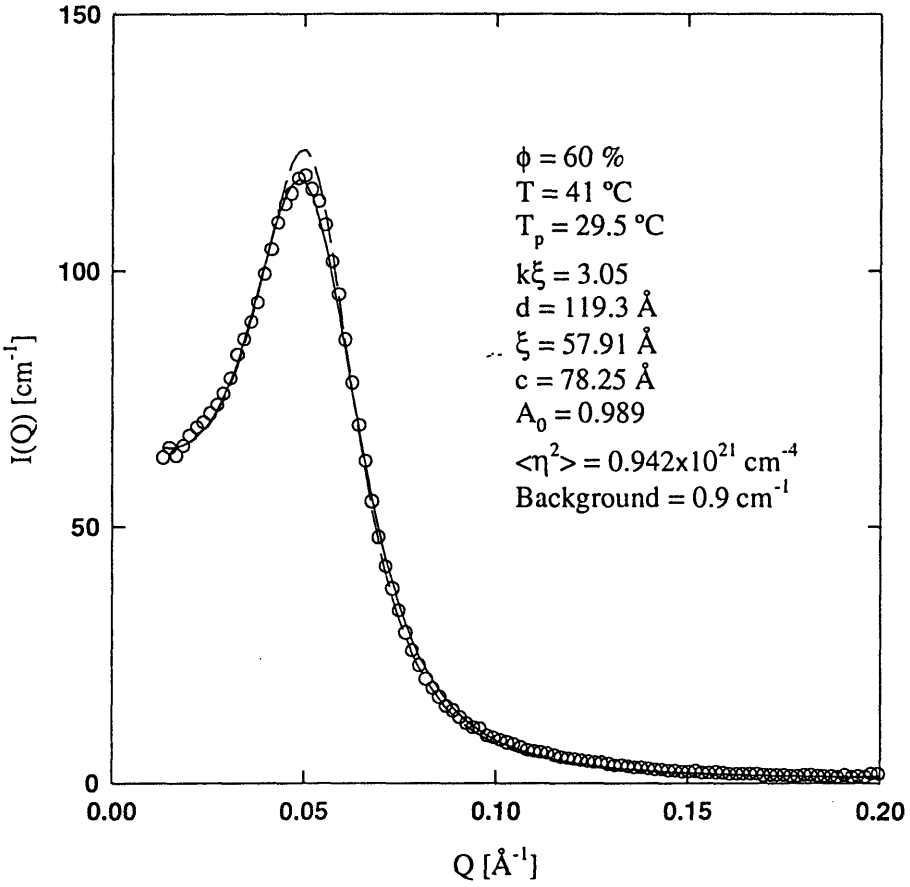

Figure 1-5: An example of including resolution function in the data analysis. Circles are experimental data, solid line is the calculation with the resolution correction. Dashed line is the calculation without the resolution correction using the same variables as shown in the figure. ϕ is the volume fraction of AOT and water, T is the temperature of the experiment, T_p is the percolation temperation, k, ξ, c, and A_0 are the fitting variables, $\langle \eta^2 \rangle$ and Background are obtained from other SANS experiments.

Chapter 2

Structure and Dynamics of Water-in-oil Microemulsions near the Critical and Percolation Points

2.1 Introduction

The three-component ionic microemulsion system made of anionic surfactant, sodium di-2-ethylhexylsulfosuccinate (AOT), water and decane is rather unusual in terms of its phase behavior. At a constant temperature, a typical symmetric ternary microemulsion system, having equal volume fractions of water and oil, shows the well-known 2-3-1 phase progression, as the surfactant concentration is increased from zero to more than 8%. When surfactant concentration is very low, the molecules are dispersed in water and oil just as monomers. The system is naturally phase separated into two-phases, with an oil- rich phase on the top and water-rich phase on the bottom because of high interfacial tension between water and oil. There is no organized structure in the two phases. At a temperature where the surfactant has balanced affinities toward water and oil, a three-phase coexistence, with a middle-phase microemulsion

in co-existence with an oil-rich phase on the top and a water-rich phase in the bottom, is to be expected at relatively higher surfactant concentrations simply because of a finite solubilization power of the surfactant for water and oil. The middle-phase microemulsion is an interesting liquid because there is an organized microstructure in it. The microemulsion shows ultra-low interfacial tensions between itself and the water and oil-rich phases. The microstructure of the middle-phase microemulsion is often described as being "bi-continuous" in both water and oil. With further increase in the surfactant concentration, a "minimum" concentration will be reached whereby all the excess water and oil are solubilized into a single-phase microemulsion. This minimum concentration is usually between 5 to 8% for a good microemulsion system. The value of the minimum concentration is a measure of amphiphilicity of the surfactant molecules at that temperature, being lower for higher amphiphilicity. In the vicinity of this minimum surfactant concentration, the microstructure of the one-phase microemulsion is also disordered bi-continuous[30]. As the surfactant concentration further increases, the one-phase microemulsion transforms into a lamellar structure, which may be called ordered bi-continuous, and then to some other three-dimensional ordered structures. This disorder-to-order transition occurs usually around 15% of the surfactant concentration.

AOT/water/decane system, on the other hand, does not follow this usual pattern of phase behavior. Fig.2-1 is the phase diagram of this system. Around room temperature the surfactant film, consisting of AOT molecules, possesses a spontaneous curvature toward water due to a hydrophilicity-lipophylicity imbalance of AOT molecules in this temperature range. Thus one finds in the ternary phase diagram a large one-phase region, called the L_2 phase, extending from the decane corner into the middle of the phase triangle. In the L_2 phase, even with equal volume fractions of water and oil, the microemulsion, instead of being bi-continuous, consists of water droplets, coated by a monolayer of AOT, dispersed in decane. With this microstructure, the microemulsion is nearly an insulator because the water droplets are separated from each other. Fig.2-2 is a schematic showing the microstructure of such a droplet. Previous Small-Angle Neutron Scattering(SANS) experiments verified that the average

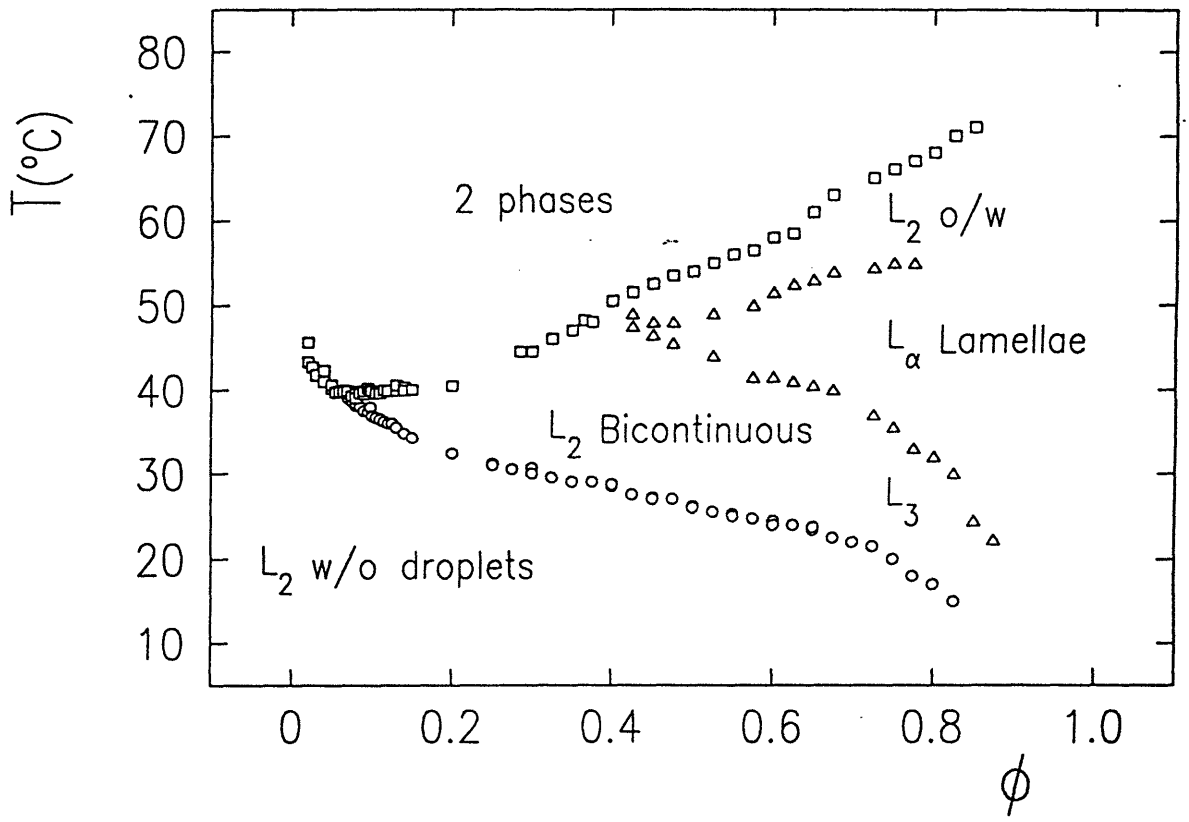

Figure 2-1: Projection of the phase prism of $AOT/H_2O/decane$ system, at $W = 40.8$ and one atmosphere pressure, on the temperature-volume fraction plane. This phase diagram shows clearly the structural transitions in the one phase region as mentioned in the text[36]. The study in this chapter concentrates on the lower left corner of this phase diagram which is shown in Fig.2-3.

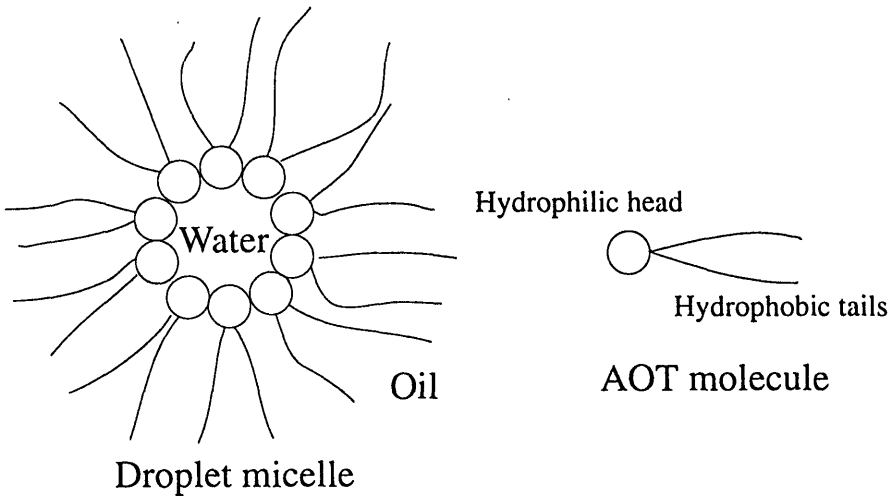

Figure 2-2: Schematic of the microstructure of a water-in-oil droplet in an AOT /water /decane microemulsion system.

radius $\langle R \rangle$ of the water droplets is determined essentially by the molar ratio of water to AOT, called W, in the system. An approximate empirical relationship between the radius (in $\mathring{A}$) and W is: $\langle R \rangle = \frac{3}{2}W$. Thus, for $W = 40.$, the average water droplet radius is about $60\mathring{A}$[31]. This water-in-oil droplet structure is maintained even if the volume fractions of water and oil are equal[32], provided the temperature is below $25°C$. This case is in sharp contrast to the usual situation that, for equal water and oil volume fractions, the microstructure of one-phase microemulsions were generally found to be bicontinuous[33, 34]. Even for the AOT/water/decane system, when a small amount of salt (NaCl) is added, the common 2-3-1 phase progression is obtained at around the hydrophile-lipophile balance temperature of $40°C$[30] and a SANS experiment in the one-phase channel at this temperature conclusively showed that the microstructure is bicontinuous[35].

This persistent droplet structure in the ternary AOT/water/decane system can, however, be used to realize an interesting coexistence of a critical phenomenon at a low volume fraction and a high temperature and a percolation phenomenon at lower temperatures but at all volume fractions. In fact, this is a rare situation in which one

has, in a real system, the realization of both the critical and percolation points at the same volume fraction.

Fig.2-3 shows the $T - \phi$ phase diagram of $AOT/H_2O/decane$ system when the water to AOT molar ratio $W = 40.8$. Substitution of H_2O by D_2O will merely shift all the phase boundaries up by about 2 degrees. ϕ denotes the volume fraction of the dispersed phase, in this case the AOT plus water. In the diagram, one sees a one- phase (L_2) region below $40°C$. In the interval of f between zero and 0.4, there is a cloud point curve separating the one-phase droplet microemulsions from two-phase droplet microemulsions. The previous SANS experiment established that the average droplet sizes and their size distributions are, within the experimental error, identical in the one-phase and two-phase regions[31]. The critical volume fraction is approximately 0.1 and the critical temperature is 40 degrees[36] in H_2O. Above the volume fraction of 0.4 there is a phase boundary between the L_2 and a lamellar phase where the microstructure is ordered and bicontinuous in water and decane (not shown in the figure).

The novelty of this phase diagram is, however, the existence of a percolation line, extending from the left of the critical point all the way to higher volume fractions, gradually decreasing in temperature to about 23 degrees at $\phi = 0.7$. Below the percolation line the microemulsion is non-conducting but above the percolation line it becomes conducting. In crossing the line, the conductivity increases by over five orders of magnitude. Fig.2-4 shows a set of logarithm of conductivities s as functions of T and ϕ[37]. One sees clearly a set of steeply rising sigmoidal curves that can be used to define a set of loci $\{T_p, \phi_p\}$ in terms of their inflection points. The asymptotic behavior of conductivity near the threshold, at a given ϕ, can be expressed as:

$$\sigma = A \cdot \left(\frac{T_p - T}{T_p}\right)^{-s'} \tag{2.1}$$

coming up from below, and

$$\sigma = B \cdot \left(\frac{T - T_p}{T_p}\right)^{t} \tag{2.2}$$

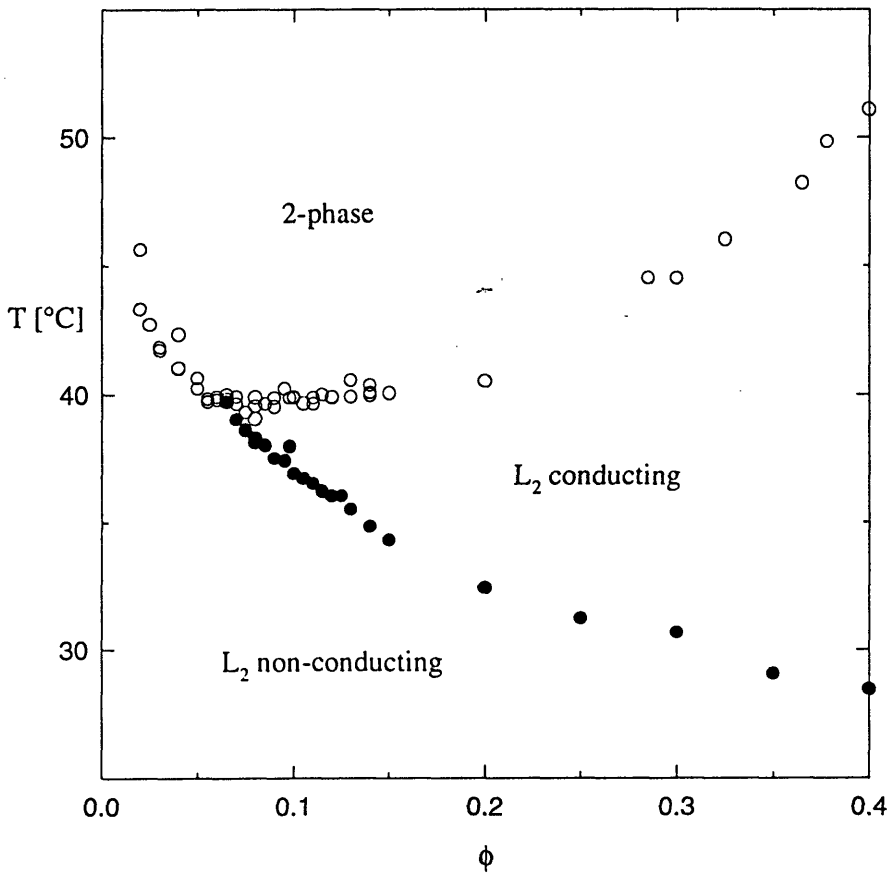

Figure 2-3: Projection of the phase prism of $AOT/H_2O/decane$ system, at $W = 40.8$ and one atmosphere pressure, on the temperature-volume fraction plane. The open circles are the one phase-two phase boundary and solid circles are percolation loci[36].

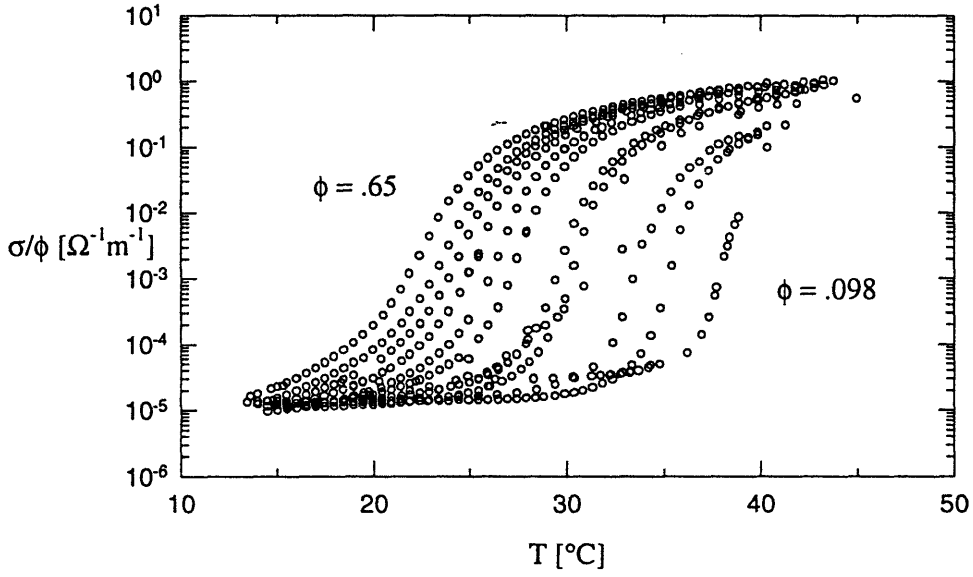

Figure 2-4: DC conductivity divided by the volume fraction of the $AOT/H_2O/decane$ system as a function of volume fraction and temperature, showing the percolation behavior[36]. Note the percolation temperature starts at about $40°C$ at 9.8% and progressively decreases to about $22°C$ at volume fraction of 65%.

going down from above. The exponents s' and t have been determined experimentally to be 1.2 ± 0.1 and 1.9 ± 0.1 respectively[37]. The exponents are the same when T is fixed but ϕ is varied[37]. The exponent s', determined from conductivities below the threshold, agrees with the value of the index proposed in the so called dynamic (or stirred) percolation theory[38, 39], which is distinct from the standard static percolation exponent $s = 0.73$[40]. On the other hand, the exponent t deduced from data above the threshold agrees with the static or geometric percolation theory[40]. In the theory of dynamic percolation, the conduction of electricity is conjectured to be mediated by charge carriers (presumably the sodium counterions from the AOT molecules) which migrate rapidly among microemulsion droplets forming transient fractal clusters, due to a short range attractive interaction between the droplets. The percolation threshold is defined theoretically to be a point where the average cluster size becomes infinity, namely, when at least one cluster spans the entire sample. Fig.2-5 is a schematic explaining this concept. Note, for this definition, finite conduction can occur already below the threshold because even there an infinite cluster can exist.

We have an experimental evidence that the percolation in AOT/water/decane system in the L_2 phase is associated with a clustering phenomenon. The phase diagram that we depict in Fig.2-3 should therefore be obtainable from the standard liquid theory with an appropriate definition of the percolation. We shall outline one such theory in the next section.

2.2 Baxter's Sticky Sphere Model and the Associated Phase Diagram

A reasonable model for a microemulsion in the L_2 phase is to regard it as a collection of spherical colloidal particles of an average radius $\langle R \rangle$ interacting with one another via a short-range temperature-dependent attractive pair-potential. This pair-potential can, for example, be a square-well potential with a hard-core diameter of a, plus an

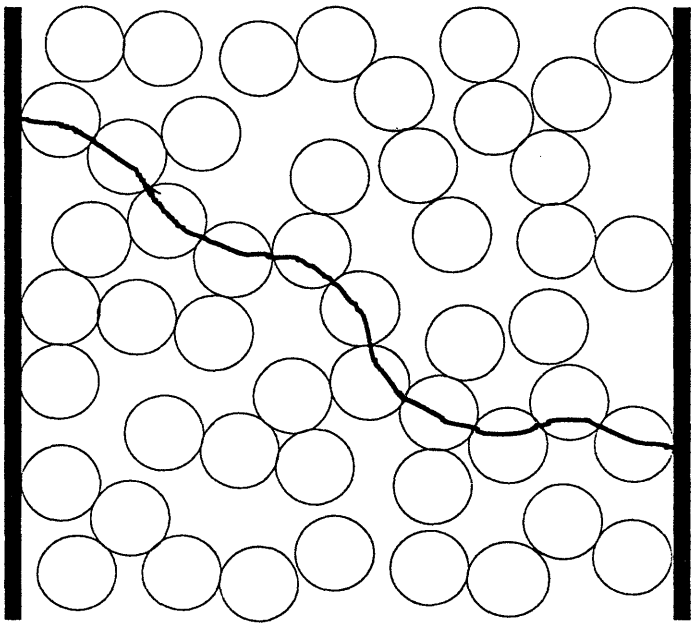

Figure 2-5: Schematic of a percolating cluster made of droplets that spans from one side to the other side.

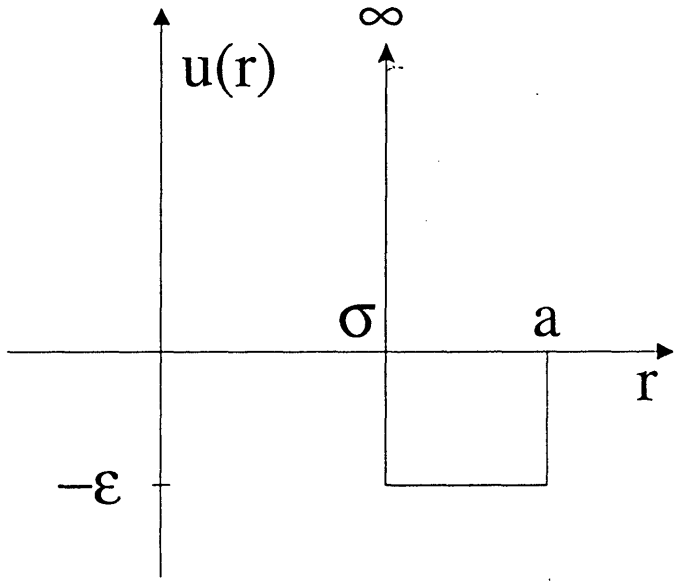

Figure 2-6: The original pair potential function used in Baxter's model.

attractive tail of depth $-\epsilon$ and width Δ as shown in Fig.2-6. The liquid theory with a square-well potential in general can not be solved in an analytical form except for a limiting case in which ϵ tends to infinity and Δ to zero in such a way that the contribution to the second virial coefficient exists. This limiting potential is called Baxter's sticky sphere potential. Specifically, the pair-potential is of the form:

$$\begin{aligned} &= +\infty && r < \sigma \\ \beta u(r) &= -\ln[(1/12\tau)(a/(a-\sigma))] && \sigma < r < a \\ &= 0 && r > a \end{aligned} \tag{2.3}$$

where $\beta = 1/k_B T$, a the outer diameter and σ the inner diameter of the attractive well. It is understood that the limit $\sigma \to a$ is to be taken in the calculation. From our discussion above, it is obvious that $a = 2\langle R \rangle$. The single dimensionless parameter $1/\tau$ is called the stickiness parameter. The sphere is stickier the smaller τ is. In the limit τ tends to infinity, the pair potential reduces to a hard sphere potential. By equating the respective second virial coefficients, one can map the square-well potential parameters into an equivalent sticky sphere potential parameter in the following way:

$$\frac{1}{\tau} = 12\frac{\Delta}{a}\exp(\beta\epsilon) \tag{2.4}$$

We assume here $a \gg \Delta$. It is seen from Eq.2.4 that the stickiness increases as $\frac{\Delta}{a}$ or $\beta\epsilon$ increase. For an AOT in decane, the parameter Δ corresponds roughly to the length of the hydrocarbon tail which can stretch out as the temperature increases.

Baxter showed that[41] the Ornstein-Zernike equation using this sticky pair- potential can be solved analytically in the Percus-Yevick approximation[42]. The PY approximation in this case amounts to a reasonable ansatz that the direct correlation function $c(r) = 0$ outside the range of the potential a. Combining this ansatz with the exact boundary condition for hard spheres that the pair-correlation function $g(r) = 0$ inside the hard core a and has a form of delta function on the surface of the sphere, the direct correlation function inside the hard core can be found. Thus one can obtain an analytical form of the three-dimensional Fourier transform of the

direct correlation function $c(k)$ as a function of the volume fraction of the spheres η and the stickiness parameter $1/\tau$. Here, $\eta = \rho a^3\pi/6$, and ρ is the number density of the particles. In comparing the theory with experiments for the scattering intensities, we shall identify η with ϕ.

First, the inter-particle structure factor $S(k)$ is calculated from the relation:

$$S(k) = \frac{1}{1 - \rho c(k)} \tag{2.5}$$

From the limiting value $S(k \to 0) = \rho k_B T \chi_T$ we can get the isothermal compressibility χ_T. By integrating χ_T with respect to the number density, one obtains the compressibility equation of states:

$$\frac{\beta p}{\rho} = \frac{1 + \eta + \eta^2}{(1-\eta)^3} - \frac{\lambda\eta(1-\eta)(1 + \eta/2) - \lambda^3\eta^2(1-\eta)^3/36}{(1-\eta)^3} \tag{2.6}$$

where the parameter λ is given by the smaller real root of:

$$\frac{\eta}{12}\lambda^2 - \left(\tau + \frac{\eta}{1+\eta}\right)\lambda + \frac{1 + \eta/2}{(1-\eta)^2} = 0 \tag{2.7}$$

From the equation of states, one finds the existence of gas-liquid phase transition with a critical point occurring at $\eta_c = 0.1213$ and $\tau_c = 0.0976$. Again, by integrating the compressibility equation of states, Barboy[43] was able to obtain an analytical chemical potential μ valid both in the one- and two-phase regions. Having the chemical potential and pressure, one can then obtain the coexistence curve by solving for the coexisting gas and liquid densities at a given τ, which is less than τ_c in the two-phase region. Fig.2-7a shows the coexistence line obtained this way in a τ_c/τ vs. η/η_c plot. In the same figure the corresponding spinodal line is also plotted in a dash line. The spinodal line is the loci of $\{\tau, \eta\}$ where the isothermal compressibility diverges.

It is seen that the coexistence curve is highly skewed toward the low volume fraction side, a feature which is often seen in micellar solutions and microemulsions. This is due to the interaction which is short range and strong and is in sharp contrast to the well-known Van der Waal's case, which is derived from an interaction that is long range and weak. To see the degree of asymmetry on the gas and liquid sides, I

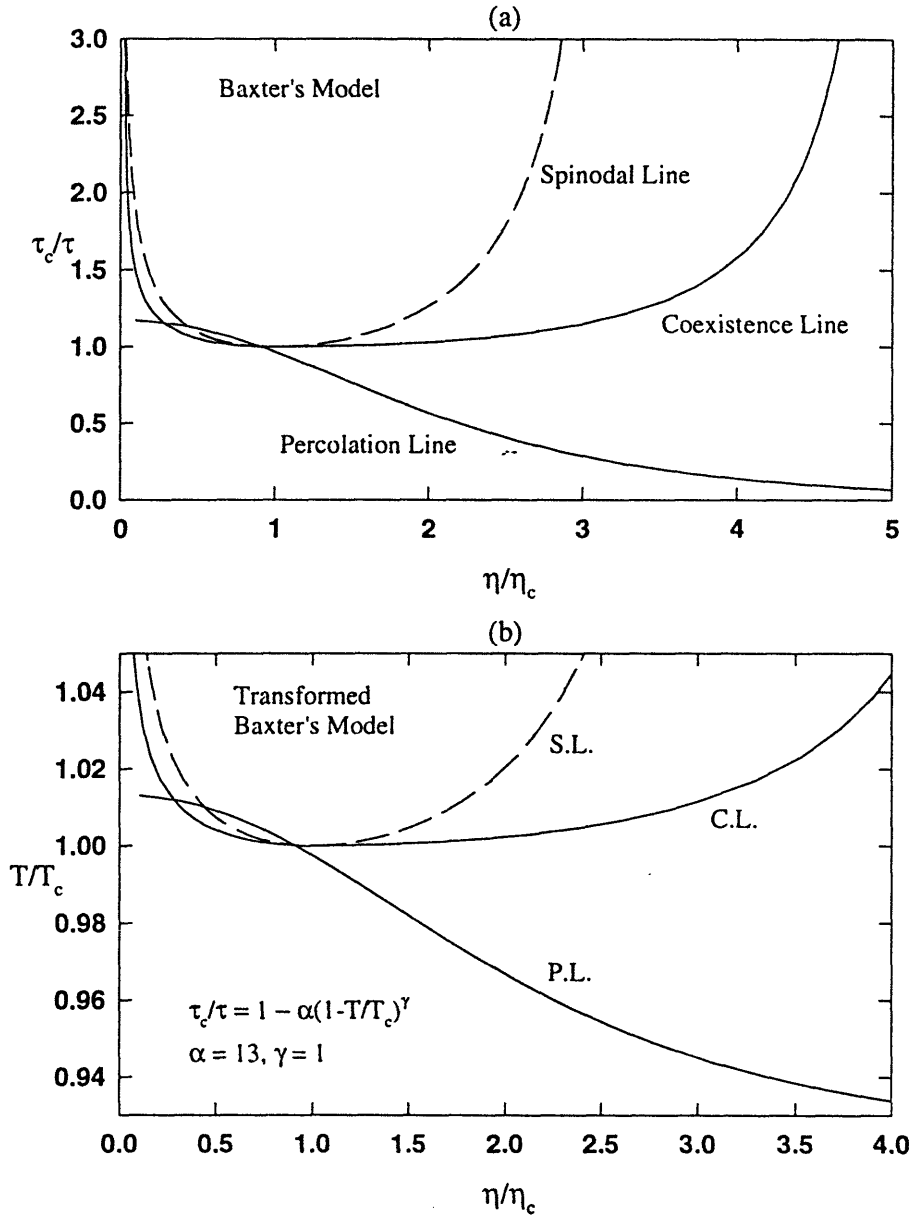

Figure 2-7: Theoretical percolation, coexistence and spinodal lines according to Baxter's model. (a) gives universal relations between τ_c/τ and η/η_c according to the model. (b) shows the results after a transformation given by Eq.2.12.

have worked out the respective asymptotic behaviors:

$$1 - \frac{\tau}{\tau_c} = 0.1584\left(1 - \frac{\eta}{\eta_c}\right)^2 \qquad \eta < \eta_c$$
$$1 - \frac{\tau}{\tau_c} = 0.0264\left(\frac{\eta}{\eta_c} - 1\right)^2 \qquad \eta > \eta_c \tag{2.8}$$

The skewness can be assessed from the ratio of amplitudes in Eq.2.8, namely $0.1584/0.0264 = 6$. This equation also shows that the PY approximation gives the mean field exponent $\beta = 1/2$.

One of the nicest features of Baxter's model is, however, that one can also derive analytically the percolation loci in the $\{\tau, \eta\}$ plane. Coniglio et al[44] introduced a pair-connectedness function $P(r)$ in 1977 in connection with development of a continuum percolation theory. Given a particle at the origin, $4\pi r^2 \rho P(r)dr$ is the number of particles in the spherical shell $(r, r + dr)$ which are connected to this central particle and belong to the same cluster. Coniglio et al showed that $P(r)$ also satisfied an Ornstein-Zernike type equation with a modified direct correlation function $c^+(r)$: By invoking the short range nature of the direct correlation function, namely, $c^+(r). = 0$, for $r > a$, and the sticky sphere condition:

$$P(r) = \frac{1}{12}\lambda a \delta(r - a) \tag{2.9}$$

Chiew and Glandt[45] was able to show that the average cluster size S is given by

$$S = \frac{1}{(1 - \lambda\eta)^2} \tag{2.10}$$

The onset of percolation can be defined as the point where S diverges. Thus percolation loci in the $\{\tau, \eta\}$ plane is given by $\eta = 1/\lambda$, leading to an equation:

$$\frac{1}{\tau} = \frac{12(1 - \eta)^2}{19\eta^2 - 2\eta + 1} \tag{2.11}$$

Fig.2-7a shows also a percolation line according to Eq.2.11.

In order to compare the theoretical phase diagram with the actual one, we have to specify the relationship between the stickiness parameter $1/\tau$ and the temperature.

Eq.2.4 suggests that $1/\tau$ is proportional to the interaction strength and the interaction strength should increase with temperature because we have a lower consolute point. The stickiness in our system is a result of entropy difference. Consider two droplets in the system as shown in Fig.2-2, when they are far apart, there are oil molecules penetrate into the hydrocarbon shell of the droplets; when they are close together, the hydrocarbon shells of them penetrate each other and expell away the oil molecules. The release of the oil molecules increases the entropy and decreases the free energy of the system so it is a favorable state for the system. At higher temperature, it is even more favorable because $U = E - TS$, where U is the free energy, E is the enthalpy, T is the temperature, and S is the entropy. Therefore, the droplets are stickier at higher temperature. The simplest relationship with two parameters α and γ is

$$\frac{\tau_c}{\tau} = 1 - \alpha\left(1 - \frac{T}{T_c}\right)^{\gamma} \tag{2.12}$$

We can try to fit the experimental coexistence curve using the sticky sphere model supplemented by Eq.2.12 and then predict the percolation loci with it. Fig.2-7b shows the results of choosing $\alpha = 13$ and $\gamma = 1$.

2.3 Analysis of SANS Data Below T_c

SANS intensity distribution from a system of polydispersed spherical droplets can be written as[46]:

$$I(Q) = (\Delta\rho)^2\phi_w\frac{4\pi}{3}\overline{R}^3\frac{(Z+6)(Z+5)(Z+4)}{(Z+1)^3}\langle\overline{P}(Q)\rangle\langle S(Q)\rangle \tag{2.13}$$

where $\Delta\rho = \rho_w - \rho_s$ is the difference of scattering length densities of D_2O and protonated decane, ϕ_w the volume fraction of D_2O , $\overline{R} = \langle R\rangle$, the average radius of the water core, and Z the index related to the polydispersity. The normalized, volume square averaged particle structure factor is defined as:

$$\langle\overline{P}(Q)\rangle = \frac{\langle R^6\left[\frac{3j_1(QR)}{QR}\right]^2\rangle}{\langle R^6\rangle} \tag{2.14}$$

The form factor of a spherical particle of radius R is $F(Q) = 3j_1(QR)/(QR)$. The form factor averaged inter-particle structure factor is defined as:

$$\langle S(Q)\rangle = \frac{\sum_{i,j}^{p}(\rho_i\rho_j)^{1/2}F_i(Q)F_j(Q)S_{ij}(Q)}{\sum_i^p \rho_i F_i^2(Q)} \tag{2.15}$$

The size average is taken with respect to a Schultz distribution, which is known to be accurate in the case of AOT/water/decane system[31]. In this case the degree of polydispersity is $\Delta R/\langle R\rangle = (1+Z)^{-1/2}$. The partial structure factor, $S_{ij}(Q)$, for a multicomponent sticky sphere system has been given by Robertus et al[47], for $i, j = 1$ to 9, using Baxter's method. The FORTRAN package for calculating the partial structure factors has been kindly supplied to us by Dr. J.G.H. Joosten. The volume square averaged particle structure factor, assuming the Schultz distribution of sizes, had previously been given in an analytical form by Kotlarchyk et al[48].

Eq.2.13 is a theory containing three adjustable parameters $\langle R\rangle$, Z and τ. These parameters are functions of temperature and volume fraction. Here we assume that particles of different sizes have the same degree of stickiness.

SANS experiments were performed at the National Institute of Standard and Technology (NIST). Wave-length of neutrons was selected at $\lambda = 5\mathring{A}$ with $\Delta\lambda/\lambda = 0.147$. The radius of the source aperture was $1.9\,cm$ and the radius of the sample aperture was $0.6\,cm$. The distance between the source aperture and the sample aperture was $1,622\,cm$. Every scattering curve was a combination of two measurements at two differnt sample-to-detector distances of $300\,cm$ and $1,110\,cm$, in order to have both low Q and high Q data.

Fig.2-8 shows results of fitting Eq.2.13 to the intensity data from samples at $\phi =$ 0.08, 0.10, and 0.12, all measured at $T = 40°C$, closest to the critical temperature. The upper three graphs show experimental $I(Q)$, in log scale, vs. Q plots and their theoretical analysis results (solid lines). The lower three graphs depicts the normalized, volume square averaged particle structure factor and the form factor averaged inter-particle structure factor extracted from the fits. The fits are satisfactory and from which we were able to extract, for all three samples, $\langle R\rangle = 50\mathring{A}$, close to what

we estimated in the introduction, and $Z = 8$, corresponding to a polydispersity index of 33%. The dimensionless parameter $\tau = 0.112$ for the 8% sample is closest to the critical value $\tau_c = 0.0976$. From this we deduce that the critical volume fraction for the microemulsion system we studied is 8%. As can be seen, the form factor averaged inter-particle structure factor shows a zero angle peak due to critical scattering and is devoid of the first diffraction peak due to the low volume fractions.

Fig.2-9 shows results of the analyses of the temperature dependence of the scattering intensity distributions from the 8% sample. As temperature increases from 30 to 35 and to 40°C the stickiness parameter progressively decreases toward the critical value, while the average size decreases and width of the size distribution increases slightly.

2.4 Analysis of the Phase Diagram

What is most pleasing to see is, however, that the temperature variation of τ_c/τ derived from the SANS data comes out in the form as given in Eq.2.12. This situation is similar to the case of non-ionic micellar solution investigated by Menon et al[49]. These authors suggested a linear relation between τ_c/τ and T/T_c. Fig.2-10 plots the τ_c/τ values of obtained from SANS data against $(1 - T/T_c)^{0.94}$. Linear relations are obtained by adjusting the value of T_c. For 8% case the T_c turns out to be $42.7°C$, close to the actual T_c in a D_2O based microemulsion system. The slope of the straight line gives $\alpha = 11.03$ in Eq.2.12.

We can derive Eq.2.12 heuristically in the following way: We study the low Q behavior of the form factor averaged structure factor $S(Q)$ for a system of sticky hard spheres of an average diameter 100Å and a polydispersity index of $Z = 10$ at the critical volume fraction $\eta_c = 0.1213$. Fig.2-11 shows that at sufficiently small Q, the Ornstein-Zernike functional form is obtained and we can thus extract the long-range correlation length ξ as a function of $1/\tau$ as we approach the critical point. Physically, as we approach the critical point, the droplets in the microemulsion form clusters. The size of the clusters become larger as the stickiness of the surfaces of

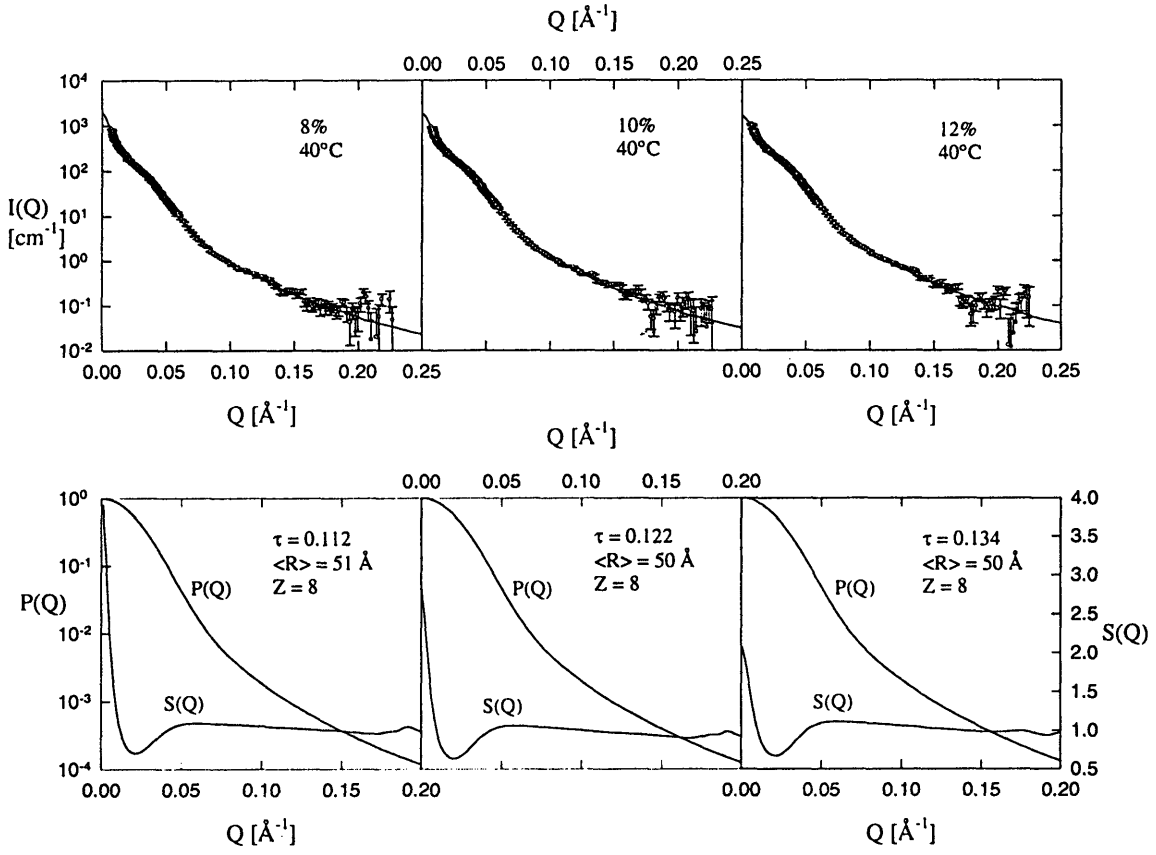

Figure 2-8: SANS intensity distributions and their analysis by Baxter's model with a polydispersity. In the top of the figure, we show SANS intensity distributions for 8, 10 and 12% volume fraction samples at 40°C. The analysis shows that 8% case is closest to the critical volume fraction as indicated by the lowest value of τ obtained. In the lower part of the figure we present the corresponding particle structure factor $P(Q)$ and the inter-particle structure factor $S(Q)$. It can be seen from the figure that $P(Q)$ for three cases are almost identical indicating the same size and size distribution of the microemulsion droplets.

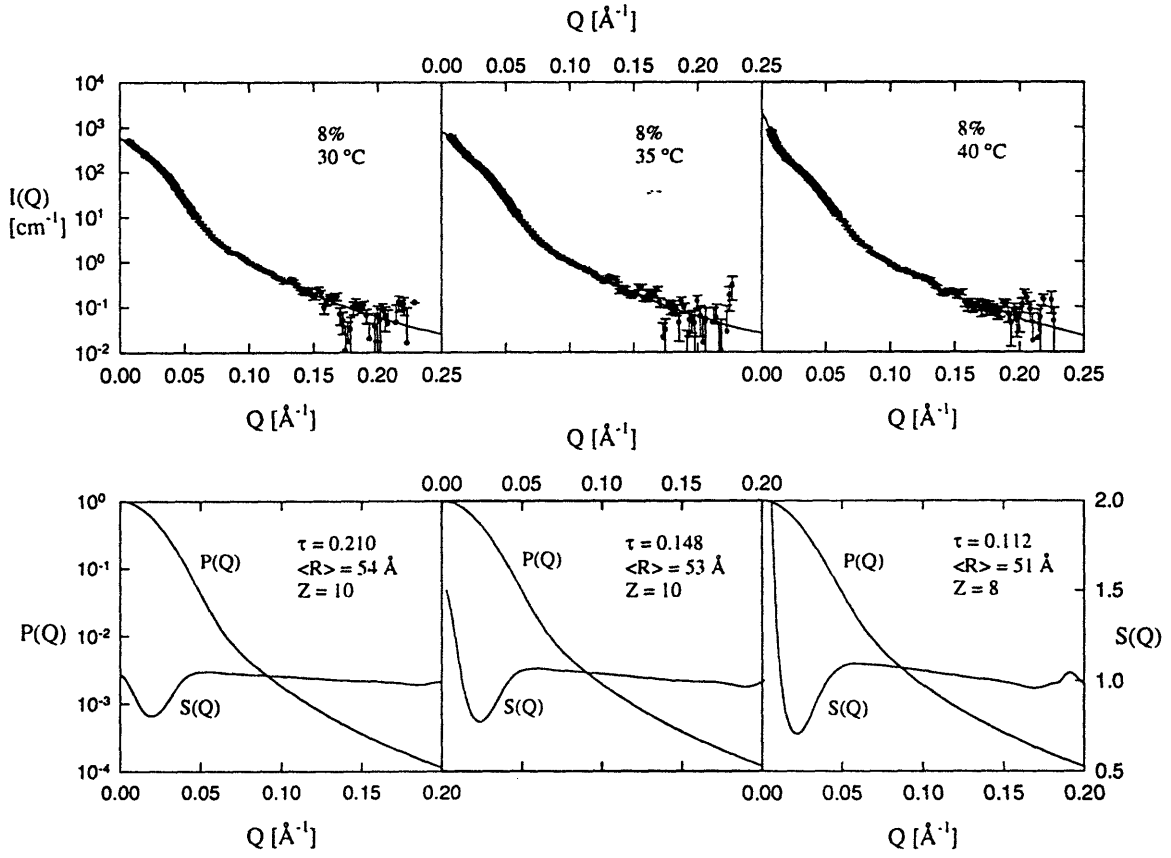

Figure 2-9: SANS intensity distributions of the 8% sample as function of temperature. It is notable that $S(Q)$ increasingly peaks in the forward direction as the temperature approaches the critical point. But the particle sizes and size distributions stay the same.

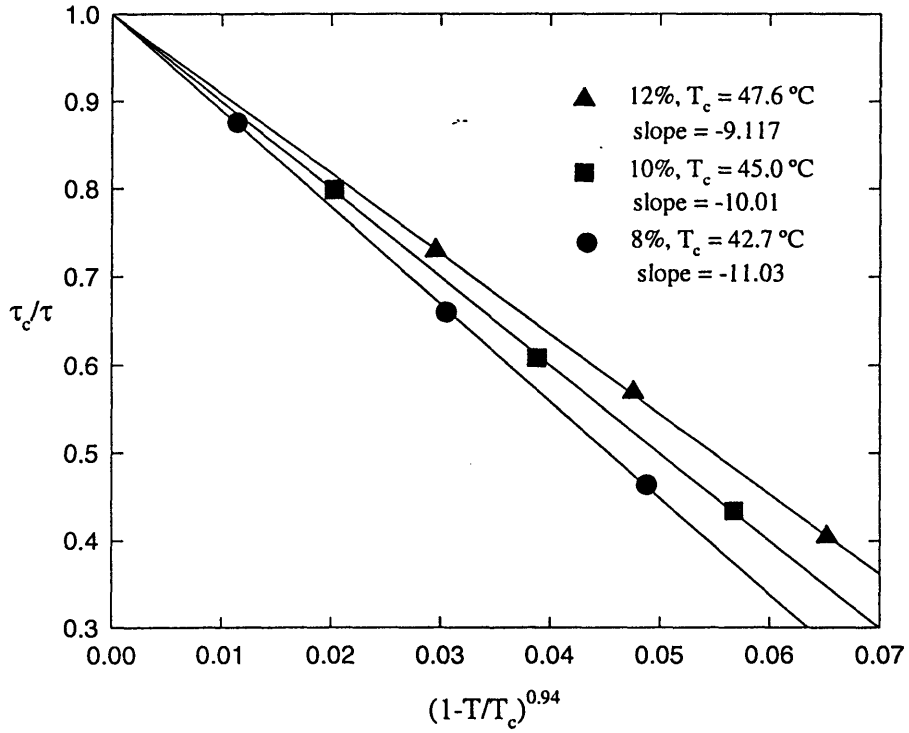

Figure 2-10: The stickiness parameter $1/\tau$ extracted from SANS data are plotted as functions of $(1 - T/T_c)^{0.94}$ in order to obtain the slope $-\alpha$ and T_c. This value of α for 8% volume fraction is used to calculate the theoretical coexistence curve shown in Fig.2-13. It is remarkable that the T_c obtained from the 8% data agree with experimental T_c measured for the AOT microemulsion made with D_2O.

the droplets increases and finally they form a separate phase. Therefore, ξ should increase as $1/\tau$ increases. Fig.2-12 plots log of the correlation length against log of $(1 - \tau_c/\tau)$. We obtain a series of straight lines implying validity of a relation

$$\xi \sim \left(1 - \frac{\tau_c}{\tau}\right)^{-\nu'} \tag{2.16}$$

where the exponent ν' depends on the polydispersity index Z. When Z is very large, namely when the system is a monodisperse sticky spheres, $\nu' = 0.5$; but when $Z = 10$, corresponding to the system under study, $\nu' = 0.532$. On the other hand, it is known experimentally as well as theoretically that near the critical point of a fluid, the correlation length is a function of the temperature distance from the critical point according to

$$\xi \sim \left(1 - \frac{T}{T_c}\right)^{-\nu} \tag{2.17}$$

with $\nu = 0.5$ as in a mean field theory like the Baxter's solution. Eq.2.16 and 2.17 taken together leads to our previous ansatz, Eq.2.12 in which

$$\gamma = \frac{\nu}{\nu'} = \frac{0.500}{0.532} = 0.94.$$

Fig.2-13 shows a comparison of experimental cloud point curve (open circles) and the theoretical co-existence curve (solid line) and spinodal line (dash line) calculated by taking a polydisperse sticky spheres with the stickiness parameter τ depending on temperature according to a relation:

$$\frac{\tau_c}{\tau} = 1 - 11\left(1 - \frac{T}{T_c}\right)^{0.94} \tag{2.18}$$

The discrepancy on the upper-right corner of Fig.2-13 is due to the microstructure of the microemulsion in that region is bicontinuous rather than droplet. The sticky-sphere model doesn't apply for a bicontinuous microemulsion.

In order to account completely for the percolation loci using Eq.2.11 and 2.12, we have to introduce a temperature dependent effective sticky sphere diameter. This

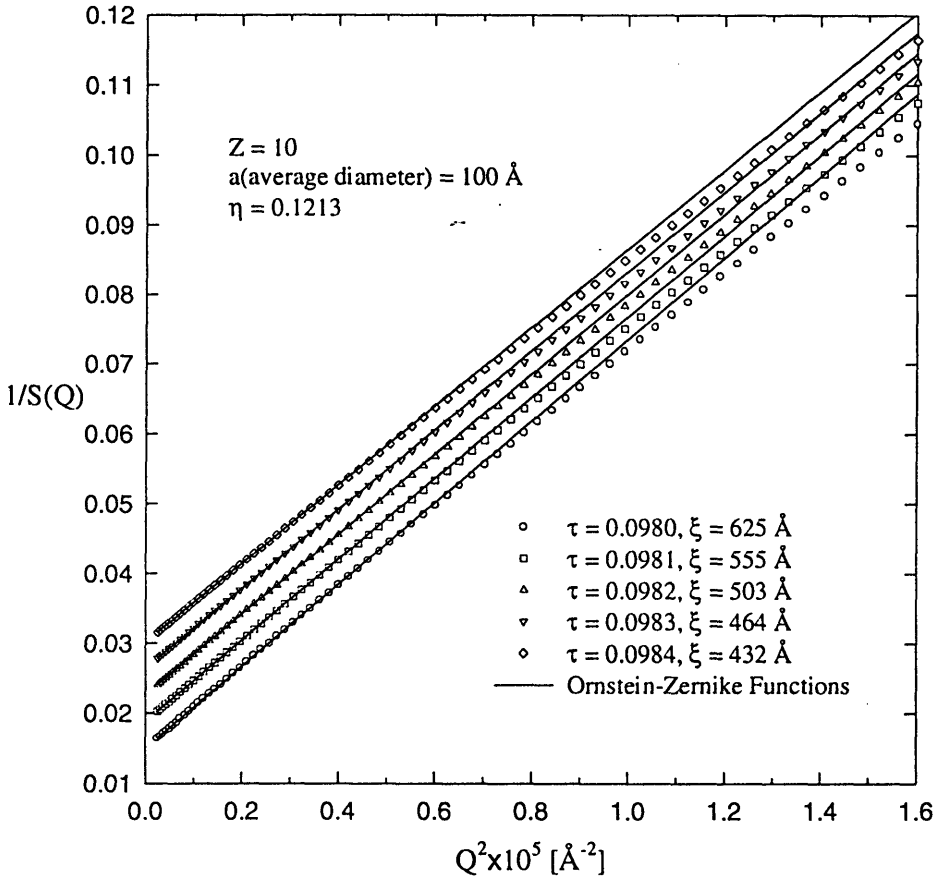

Figure 2-11: Small Q behavior of $S(Q)$ function near the critical point. The graph shows that $S(Q)$ follows the Ornstein-Zernike function at sufficiently small Q so that the correlation length ξ can be extracted. It can be seen that as τ approaches $\tau_c(= 0.0976)$ the correlation length increases.

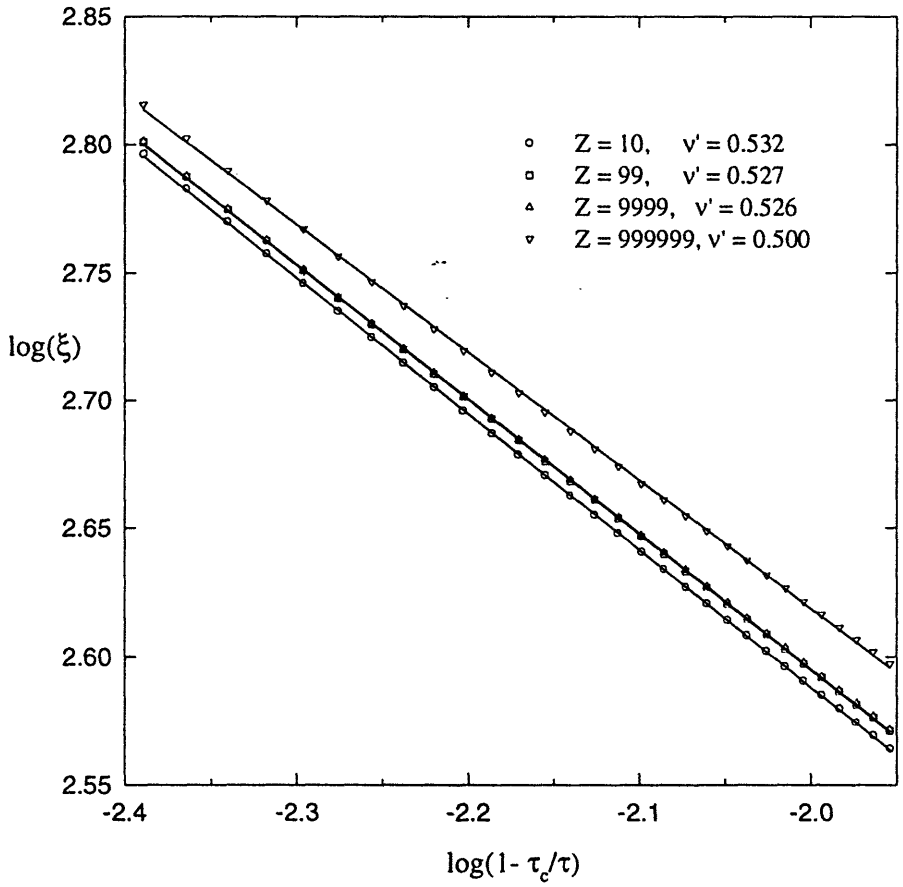

Figure 2-12: The dependence of the extracted correlation length as a function of $(1 - \tau_c/\tau)$. It is seen that near the critical point, this is a power law dependence with an index ν'. For a monodisperse system (corresponding to large Z), ν' is 0.5. As the polydispersity increases (corresponding to decreasing value of Z), the value of ν' increases. At 33% polydispersity, ν' is 0.532.

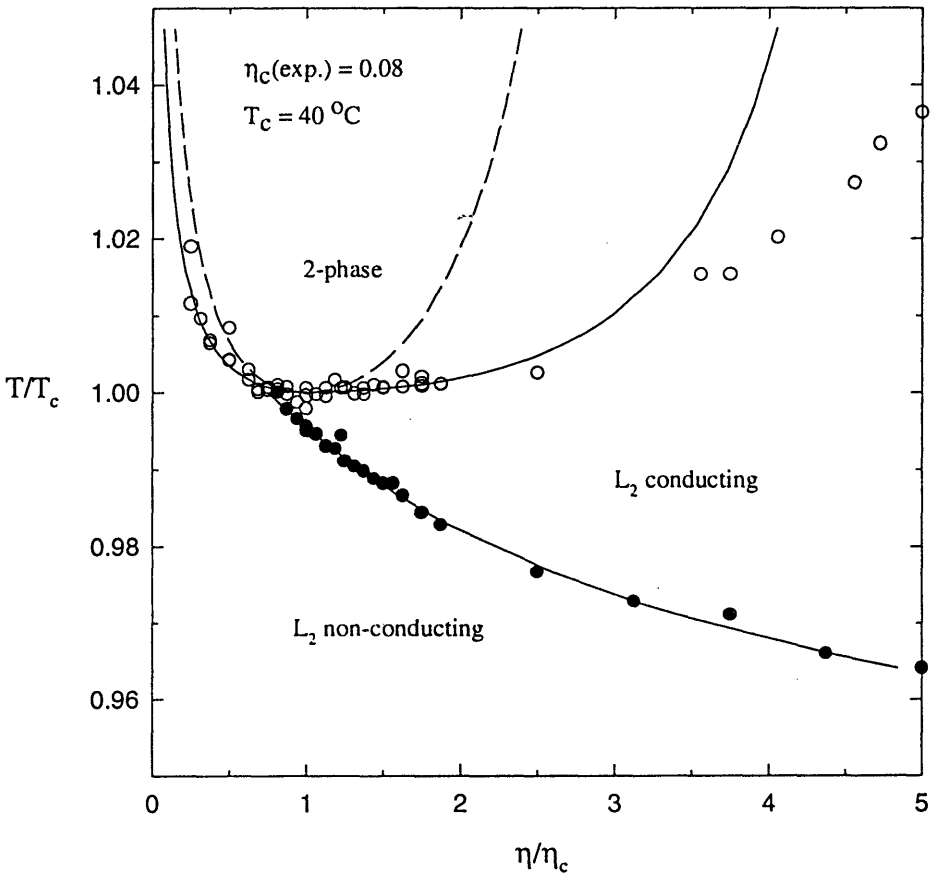

Figure 2-13: Experimental cloud points and the corresponding theoretical coexistence and spinodal curves based on Baxter's model after the transformation of Eq.2.12, with $\alpha = 11$ and $\gamma = 0.94$ (see text). The percolation loci was fitted with the Baxter's model using the method described in next figure caption.

idea is reasonable because the definition of connectivity of two spheres should be dependent on thermodynamic state of the liquid. In fact it is intuitively appealing to postulate that the higher the temperature the easier for the counterions to migrate from one water core of a droplet to another in the neighborhood. Therefore the effective diameter of the microemulsion droplets, as far as electrical percolation is concerned, is larger for higher temperatures. Fig.2-14 shows the result of force fitting the experimental percolation loci (solid circles) with Eq.2.11 (solid line). The ratio, $(\eta_{SHS}/\eta)^{1/3}$, which is a measure of the ratio of the effective diameter to the actual diameter, turns out to be a linear function of T/T_c. It is shown in Fig.2-14 as a solid line.

2.5 Dynamics of the Droplet Number Density Fluctuation near the Critical Point

We shall turn next to the discussion of some aspects of the droplet dynamics near the critical point. The starting point of our theory is an assumption that the slow dynamics of the droplets is dominated by diffusive motions of the percolation clusters[50]. This assumption is expected to be good in the vicinity of the percolation threshold where large, transient fractal clusters are formed. Formation of the transient fractal clusters is a necessary condition for the dynamic percolation theory[38] to be valid. We have used it to explain the conductivity exponent below the percolation threshold in the introduction. In AOT/water/decane system, as one can see from the phase diagram (Fig.2-3), the critical point is only about two degrees above the percolation point. One therefore expects that the cluster structure and cluster size distribution are similar in the critical region to the percolation point.

For light scattering, the wave-length of visible light is much larger than the droplet sizes. Hence, for this Q range, the particle structure factor is nearly unity and we can ignore it.

First, we calculate the inter-particle structure factor $S_k(Q)$ for a cluster containing

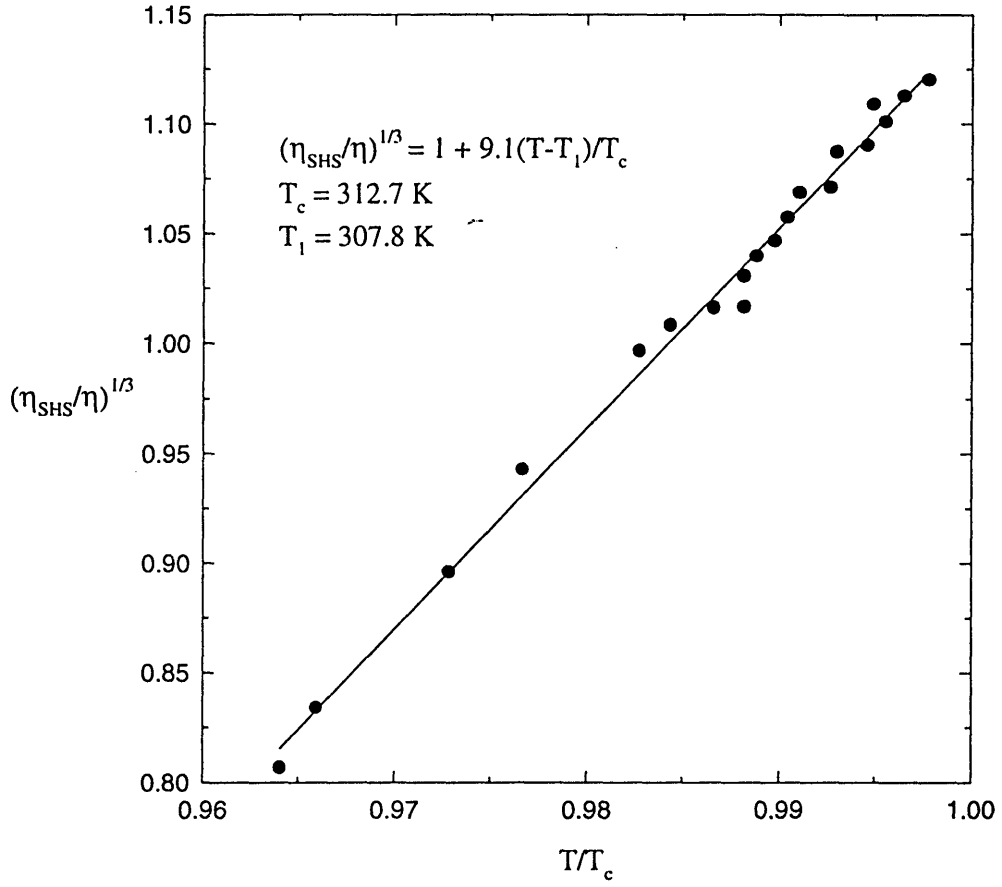

Figure 2-14: The temperature dependence of the apparent volume fraction η_{SHS} of sticky spheres which are needed to fit the percolation line. We plot the ratio $(\eta_{SHS}/\eta)^{1/3}$ versus T/T_c which results in a straight line. This shows that the apparent diameter of sticky spheres which is percolating increases as the temperature increases.

k particles. We do this by Fourier transforming the k-cluster pair correlation function,

$$\rho(g_k(r) - 1) = \frac{D}{4\pi R_1^D} \frac{e^{-r/R_k}}{r^{3-D}} \tag{2.19}$$

where D is the fractal dimension of the clusters, $R_k = R_1 k^{1/D}$, the radius of gyration of the k-cluster, and R_1 the average radius of the droplet. The result can be put in an analytical form[51]:

$$S_k(Q) = \frac{kD\Gamma(D)\sin[(D-1)\arctan(QR_k)]}{(D-1)QR_k[1+Q^2R_k^2]^{(D-1)/2}} + 1 \tag{2.20}$$

With the structure factor, the intermediate scattering function $F(Q,t)$ can be calculated as

$$F(Q,t) = \frac{\sum_{k=1}^{\infty} kN(k)S_k(Q)\exp(-D_kQ^2t)}{\sum_{k=1}^{\infty} kN(k)S_k(Q)} \tag{2.21}$$

The discrete sum can be converted into an integral over k by introducing a cluster size distribution function of the form: $N(k) \approx k^{1-\tau}\exp(-k/S)$[52]. In this expression τ is the polydispersity exponent, S the average cluster size, $D_k = D_1 k^{-1/D}$ the translational diffusion coefficient of the k-cluster and D_1 the Stokes-Einstein diffusion coefficient of the droplet. Numerical simulations for three-dimensional percolation clusters gave a fractal dimension $D = 2.5$ and the polydispersity exponent $\tau = 2.2$[52]. The measured photon correlation function is then given by $C^2(Q,t) = 1 + |F(Q,t)|^2$.

The first cumulant, or the average relaxation rate, is the logarithmic derivative of $C(Q,t)$ evaluated at $t = 0$. It is given by:

$$\Gamma(Q) = \frac{\sum_{k=1}^{\infty} kN(k)S_k(Q)D_kQ^2}{\sum_{k=1}^{\infty} kN(k)S_k(Q)} \tag{2.22}$$

2.5.1 Dynamic Slowing-Down of the Average Relaxation Rate

Eq.2.22 for the average relaxation rate $\Gamma(Q)$ can be expressed in terms of two dimensionless variables $x = Q\xi$ and $x_1 = QR_1$, where the correlation length is defined

as $\xi = R_1 S^{1/D}/3^{1/2}$. The complete analytical form of the $\Gamma(Q)$ near the percolation threshold has been given in [50]. We shall limit our discussion here to the particular case near the critical point. In this case the cluster structure factor Eq.2.20 is approximated by its low Q form $S_k(Q) = k\exp(-Q^2R_k^2/3)$. An analytical expression of $\Gamma(Q)$ is obtained by converting Eq.2.22 into an integral. It is more revealing to construct a scaling function defined as: $\Gamma^*(x, x_1) = \Gamma(Q)/D_1R_1Q^3$. It is given by [53]

$$\Gamma^*(x, x_1) = \frac{3\pi}{8}\frac{\Gamma(3-\tau, x_1^D)\Gamma(3-\tau-1/d, u)}{\Gamma(3-\tau-1/D, x_1^D)\Gamma(3-\tau, u)}\left[1+\frac{1}{x^2}\right]^{1/2} \tag{2.23}$$

where $u = \left(\frac{x_1}{x}\right)^D [1+x^2]^{D/2}$, and $\Gamma(a, b)$ is the incomplete Euler gamma function. It should be remarked that the presence of the second non-universal variable x_1 in Eq.2.23 is due to the finite size of the constituent particles. In this regard, it is remarkable that in the limit of small particles,

$$\lim_{x_1 \to 0} \Gamma^*(x, x_1) \approx \frac{K(x)}{x^3} \tag{2.24}$$

where $K(x)$ is the Kawasaki universal dynamic scaling function originated from a mode coupling theory:

$$K(x) = \frac{3}{4}\left[1+x^2+\left(x^3-\frac{1}{x}\right)\arctan(x)\right] \tag{2.25}$$

which is known to account for light scattering data from binary mixtures of molecular liquids very well. $\Gamma^*(x, x_1 = 0)$ has simple asymptotic behavior

$$\begin{aligned} &\text{For } x \ll 1 \qquad \Gamma^*(x) = a/x \\ &\text{For } x \gg 1 \qquad \Gamma^*(x) = b \end{aligned} \tag{2.26}$$

where a and b are known constants. Fig.2-15 illustrates the cross-over from the small x to large x behavior as expressed by Eq.2.26. Using light scattering data taken near the critical point of the AOT/water/decane system, we illustrate the agreement of measured first cumulants of photon correlation functions and prediction of Eq.2.23[53]. It is clear from the graph that the finite size effect of microemulsion

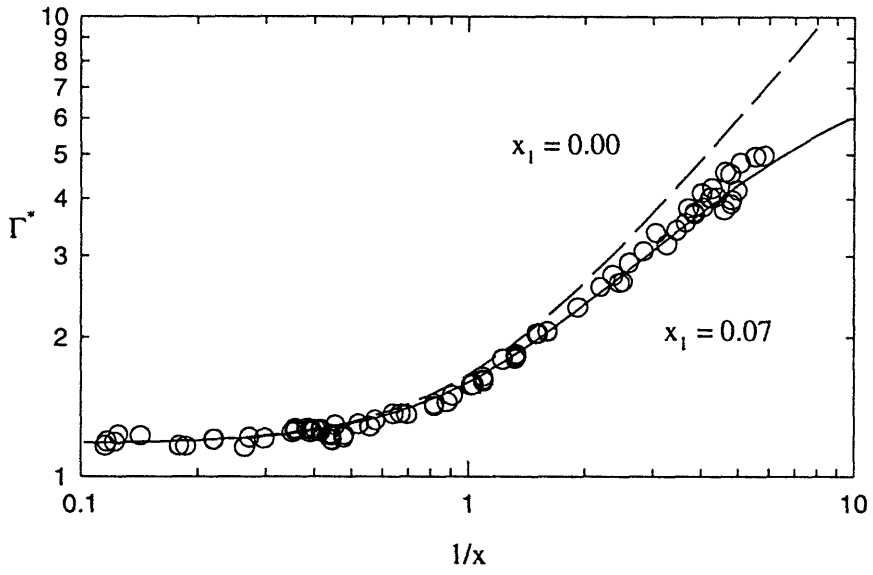

Figure 2-15: The dynamic scaling function $\Gamma^*(x, x_1)$ associated the average relaxation time of the cluster diffusion plotted as a function of $1/x$ for two values of the scaled droplet size x_1. Open circles are experimental data from AOT/water/decane system near the critical point. The dashed line corresponds to Kawasaki's mode-mode coupling result. The solid line is the dynamic droplet model result presented in the text.

droplets is large enough to be detectable in a light scattering experiment.

Analogous dynamic slowing-down phenomenon near percolation points at higher volume fractions has been observed and explained by the theory [50, 54].

2.5.2 Stretched Exponential Decay of the Time Correlation Function at Long Time

Again Eq.2.21 in its integral form can be computed analytically[55]. We then obtain the time correlation function, $C(u, v)$, in terms of the scaling variable u and a dimensionless time variable $v = D_1 R_1 Q^3 t(1 + x^{-2})^{1/2}$. It is sufficient for the purpose here to state that at sufficiently short time, we have an exponential decay with the average relaxation rate $\Gamma(Q)$ given before,

$$C(u, v \to 1) = \exp[-\Gamma(Q)t] \tag{2.27}$$

and at long time such that $\Gamma t \gg 1$, the time correlation function approaches an stretched exponential form,

$$C(u, v \gg 1) = \exp[-(\overline{\Gamma} t)^{\beta}] \tag{2.28}$$

with $\overline{\Gamma}$ given by

$$\overline{\Gamma} = \beta^{1/\beta} D^{1/D} D_1 R_1 q^3 \left(1 + \frac{1}{x^2}\right)^{1/2} \tag{2.29}$$

and the exponent $\beta = D/(D+1)$ is a universal number 0.713.

Cross-over from the short-time exponential decay to the long-time stretched exponential decay of the photon correlation function occurs at the dimensionless variable $\Gamma t = 1$. Near the percolation points at high volume fractions, one has a similar cross-overs from exponential to stretched exponential behavior[50], except the cross-overs occur at earlier times so the stretched exponential decays can be easily observed in experiments[46].

2.6 Conclusion

We have given concrete evidence that both the structure and dynamics observed in a three-component microemulsion system, AOT/water/decane, near the critical and percolation points can be explained in terms of a model based on the formation of transient, polydisperse fractal clusters due to a short-range attraction between microemulsion droplets. This attractive interaction increases in a specific way as the temperature increases toward the percolation and critical points. We derive a quantitative relation between the interaction strength and the temperature from analyses of SANS data in the one-phase region approaching the critical point. This relation serves to explain the over-all feature of the phase diagram, including the cloud-point curve and the percolation line.

The diffusive cluster dynamics also accounts for the Q-dependence of the first cumulant of photon correlation functions in the critical region. The dynamic scaling

function associated with the average relaxation rate deviates significantly from the well-known Kawasaki function at large Q due to the large sizes of the microemulsion droplets[53]. The long-time behavior of the photon correlation function is shown to be a stretched exponential form with a universal stretch exponent of 0.713[50].

A FORTRAN computer program used to calculate the structure factor and interparticle structure factor for a multicomponent droplet system using Baxter's model is listed in Appendix B.

Chapter 3

Structural Study of the Liquid and Gel Phases of a Semifluorinated Alkanes in a Mixed Solvent

3.1 Introduction

Fluorocarbons and hydrocarbons are both hydrophobic and apolar molecules, with very low dipolar moments, therefore in principle they are expected to form regular mixtures, due to the weakness of the intermolecular forces. Instead, binary mixtures of C_mF_{2m+2} and C_nH_{2n+2} are an interesting and surprising example of phase separating systems[56, 57, 58].

The plot of demixing temperature(T_0) versus concentration(x) provides the phase separation curve with an upper critical solution temperature(UCST) and a critical concentration value x_c. The curve is quite asymmetric and the values of x_c and T_c depend on the chemical structure of the two liquids.

Unlike other phase separating systems, mixtures of fluorocarbons and hydrogenated solvents show larger deviations from the "regular solution" theory (S2,M2) than those expected from the difference in solubility parameter δ.

In spite of their chemical similarity, fluorocarbons and hydrocarbons possess dif-

ferent properties and are immiscible below a certain demixing temperature, as a result of the different chain conformations: in fact $(CH_2)_n$ segments arrange in the usual zigzag conformation with a cross-section area of 18.5 $\AA^2$/molecule, whereas a $(CF_2)_m$ chains possesses a typical 15/7 helix conformation and the cross-section area is 28.3 $\AA^2$/molecule[60]. For this reason fluorocarbon/hydrocarbon mixtures show characteristic phase boundaries (demixing temperature versus either molar or volume fraction) with a broad maximum that corresponds to the upper critical solution temperature (T_c) and to the critical concentration (x_c) or critical volume fraction (ϕ_c).

Semifluorinated n-alkanes (or perfluoroalkyl-alkanes) are diblock short-chain copolymers that show peculiar properties in the liquid and in the solid state. They are constituted by two different blocks: a fluorinated chain linked to a hydrocarbon tail $F_3C(CF_2)_{m-1}-(CH_2)_{n-1}CH_3$ (shortly F_mH_n). Since the two segments are covalently bonded, they cannot phase separate.

The presence of both fluorinated and hydrogenated tails in a semifluorinated alkane produces interesting phenomena both in the solid and in the liquid state. In the solid state F_mH_n produce liquid crystals, show peculiar ordered arrangements, and undergo some phase transitions that have been extensively studied[62, 63, 64, 65, 66, 67, 68, 69].

In the liquid state, semifluorinated alkanes produce micellar aggregates when dissolved in fluorinated or hydrogenated solvents[70, 71, 72, 64]. Being completely insoluble in water, they also form stable Langmuir monomolecular films and show a slight surface activity at the air/water interface[74]. Heating up a solution containing F_mH_n micellar aggregates results in the destruction of these aggregates, whilst at lower temperatures the solution forms a gel, depending on the nature of the solvent and on the concentration of the copolymer. These gels are constituted by long fibers of copolymer in an extended network that entraps free solvent molecules. The gels look like soft, high viscosity materials, that partially dissolve by shaking. By heating up the gel, a clear solution is quickly restored.

F_mH_n solutions in different solvents have been usually studied by several tech-

niques such as static and dynamic light-scattering, NMR, UV absorption, small-angle neutron and X-ray scattering (SANS and SAXS respectively), differential scanning calorimetry (DSC), surface tension, and dye solubilization measurements. These data clearly show that the copolymer monomers form micellar aggregates, depending on the temperature, on the chemical composition of the F_mH_n and on the nature of the solvent. The incompatibility between fluorinated and hydrogenated blocks is the main factor that controls the formation of such aggregates. In fact in the presence of a selective solvent, the two blocks will establish different interactions with the solvent molecules and therefore will be confined in different regions of the supramolecular structure.

Since F_mH_n copolymers are able to form aggregates when dissolved in a selective solvent, they have been named as "primitive amphiphiles"; this term refers to an apolar molecule that is formed by two incompatible units and possesses a low dipolar moment. According to this recent definition, primitive amphiphiles represent the ancestors of the whole amphiphile' family, because of the presence in the same molecule of two incompatible and immiscible moieties, and therefore it is expected that they will provide a useful tool for testing molecular theories of surfactant aggregation.

The addition of even small quantities of F_8H_{16} to a mixture of perfluorooctane (PFO) and i-octane (i-OCT) produced a significant lowering of the upper consolute temperature and a relevant broadening of the phase separation curve. Light-scattering from the liquid sample was not ascribed to the presence of small aggregates, but to a critical scattering related to the demixing of the two solvents.

Larger amounts of copolymer added to the PFO/i-OCT system resulted in the formation of a white gel below a certain liquid-gel transition temperature, and in this case no solvent demixing was observed upon cooling. In this high-copolymer state, dynamic light-scattering and SAXS measurements performed on the liquid samples showed the presence of small aggregates characterized by an average hydrodynamic diameter of about 30 Å. Lowering the temperature, the system evolves toward the formation of a bigger structure and eventually to the formation of an extended, ribbon-like, sheet.

According to the model for the liquid phase[60], some copolymer molecules aggregate in the mixed solvent and form small micelles. As the temperature decreases and approaches the liquid-gel phase transition temperature, the original micelles grow and form an extended, ribbon-like structure, where the copolymer molecules produce a lamellar pattern, with the fluorinated chains closely packed in a side-by-side and head-by-head arrangement. The hydrogenated segments will interdigitate in the internal region, given the smaller cross-section of the hydrocarbon blocks respect to the section of the fluorinated helices. As the temperature lowers, other copolymer monomers will be captured and several lamellar layers will be formed, producing birefringent elongated structures that can be observed under a polarizing microscope.

Two plausible models are proposed for the gel phase based on the birefringence picture[60, 70] which includes both the fine structure of the crystalline part and the mesoscopic aggregational behavior that leads to the formation of the fibers. We then construct the particle structure factors which approximately describe both the internal structure and the mesoscopic structure of the fibers.

3.2 Materials and Methods

Iso-octane was purchased from Fluka (Buch, Switzerland) and perfluorooctane (98%) was supplied by M&G Chemicals (Stockport, England). The solvents were used without further treatments. The semifluorinated n-alkane F_8H_{16} was obtained according to the two-step procedure already described previously[72]:

1. $F(CF_2)_8I + CH_2{=}CH\text{-}(CH_2)_{14}H \longrightarrow F(CF_2)_8\text{-}CH_2\text{-}CHI\text{-}(CH_2)_{14}H$

2. $F(CF_2)_8\text{-}CH_2\text{-}CHI\text{-}(CH_2)_{14}H + H_2 \longrightarrow F(CF_2)_8\text{-}\ (CH_2)_{16}H$

The synthesis was performed by Dr. P. Lo Nostro at University of Florence, Italy. F_8H_{16}, PFO, and i-OCT were weighted directly in a glass tube and the sample was slowly heated in a thermostated bath(± 0.1°C), until it became completely clear, cooled down to phase separation, and then heated up again. The cooling-heating cycle was repeated slowly and several times on each sample. The liquid-liquid phase

separation temperature(T_0) was then calculated as the average of the cloud point temperature(T_{cp}) and the reclarification temperature (T_{re}). The difference between T_{cp} and T_{re} was always within ± 0.5°C. The sample composition was then changed, and the phase separation curve obtained by plotting T_0 as a function of PFO molar fraction(x_1).

For larger amounts of copolymer, the homogeneous liquid mixture does not phase separate into two liquid phases upon cooling, but rather forms a white solid gel. The liquid-gel transition temperature(T_g) can be measured by recording the temperature at which the gel first appears in the liquid mixture when the sample is cooled down(T_1), and the temperature at which the gel converts into a clear liquid upon heating(T_2). T_g is then calculated as the average between T_1 and T_2.

Density, refractive index and viscosity measurements were carried out following the procedures reported in [72]. Birefringence observations were made with a Reichert-Zetopan polarizing microscope, with crossed nicols, on the liquid and gel samples. Static and dynamic light-scattering measurements were performed with a Brookhaven Instr. Co. apparatus($\lambda = 514.5\,nm, Ar^+$ source) and the data analyzed with a BI-2030 AT Digital Correlator. The scattering intensity was recorded at different temperatures and at different scattering angles ($15° < \theta < 160°$). Samples for light-scattering experiments were filtered in order to remove the dust.

Small-angle X-ray-scattering experiments were carried out at the Small-Angle X-Ray-Scattering Facility, Oak Ridge National Laboratory, Oak Ridge, TN. The spectrometer is equipped with a rotating anode X-ray source and a $20 \times 20\ cm^2$ position-sensitive area detector with 64×64 pixels. X-Rays from Cu K_α line with $\lambda = 1.54\mathring{A}$ were extracted using a graphite monochromator. The source-to-sample distance was 2 m, with three collimation pin holes, the resulting beam spot at the sample position is 1.5 mm. The sample-to-detector distance was 2.065 m. This configuration gave us a Q range of $0.01 - 0.25\mathring{A}^{-1}$. The scattering data were corrected for background, detector efficiency, and sample cell scattering. The absolute normalization of the scattering intensity was made with a vitreous carbon standard.

Table 3.1: List of data extracted from the phase separation curves.

	$T_c(K)$	x_c	$(\phi_{PFO}/\phi_{i-OCT})_c$
$X = \infty$ (curve A)	300.4	0.321	0.71
$X = 20$ (curve B)	290.4	0.422	1.10
$X = 10$ (curve C)	283.5	0.472	1.36

3.3 Phase Diagram, Light Scattering, and Birefringence

Fig.3-1(a) shows the liquid-liquid phase separation temperature(T_0) as a function of the PFO molar fraction(x_1), for different $PFO/i-OCT/F_8H16$ mixtures. Defining X as the molar ratio between i-OCT and F_8H_{16}, curve A corresponds to $X = \infty$(no F_8H_{16} in the mixture), while curves B and C correspond to $X = 20$ and $X = 10$ respectively. In all cases the phase separation curve shows an upper consolute temperature T_c for a critical molar fraction x_c. The significant asymmetry of the curves accounts for the large differences in molecular sizes between PFO and i-OCT[73, 74]. The skewness of the curve and the value of T_c depend on the amount of copolymer added to the PFO/i-OCT system, showing that the critical demixing is greatly affected by the presence of the semifluorinated n-alkane. In Table 3.1 we summarize the experimental parameters obtained from the coexistence curves of the $PFO/i-OCT/F_8H_{16}$ mixtures. $(\phi_{PFO}/\phi_{i-OCT})_c$ represents the volume fraction ratio of PFO over i-OCT at the critical point.

The effect induced by the semifluorinated compound on the PFO/i-OCT system is indicated by the relevant decrement of T_c as more F_8H_{16} is added to the mixture, whereas x_c and ϕ_{PFO}/ϕ_{i-OCT} change only in a slight way as a function of X.

The data reported in Table 3.1 indicate that the presence of F_8H_{16} makes the PFO/i-OCT mixture more stable, increasing the amount of PFO that can be added before the critical phenomenon occurs; in other words the copolymer keeps the mixture in the monophase region, whereas at the same composition and temperature the PFO/i-OCT system would phase separate. F_8H_{16} acts then as a "compatibilizing" agent for the PFO/i-OCT system, decreasing the incompatibility of the two solvents.

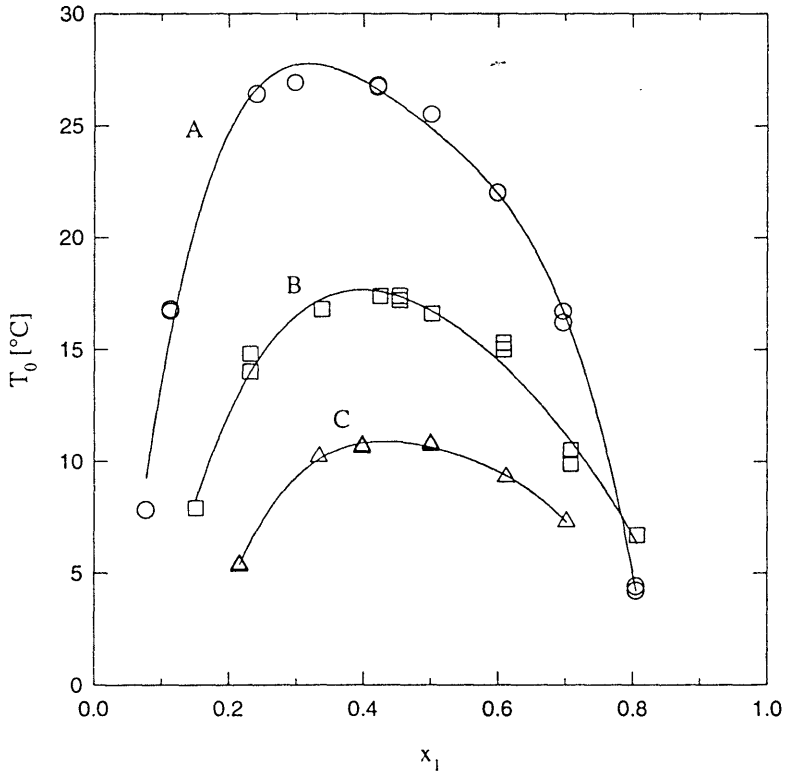

Figure 3-1: Phase separation curves for different $PFO/i - OCT/F_8H_{16}$ mixtures: curve A, no copolymer($X = \infty$); curve B, $X = 20$; curve C, $X = 10$.

This "compatibilizing" effect reproduces the behavior shown by other semifluorinated copolymers with different solvents[70].

In order to explain the phase separation, the Hildebrand-Scatchard equation, which is valid for mixtures of fluorocarbons and linear hydrocarbons, can be applied to the PFO/i-OCT system (curve A). This equation relates the critical temperature to the molar volume and to the solubility parameter of each component[75]:

$$RT_c \approx (\delta_1 - \delta_2)^2 \frac{2V_1V_2}{(\sqrt{V_1} + \sqrt{V_2})^2} \tag{3.1}$$

where R is the gas constant ($1.9872\, cal \cdot mol^{-1} \cdot K^{-1}$), V_i and δ_i are the molar volume and solubility parameter of each component respectively; δ is defined as:

$$\delta_i^2 = \frac{\Delta H_i^{vap} - RT}{V_i} \tag{3.2}$$

where ΔH_i^{vap} is the molar enthalpy of vaporization of component i. According to our experiments, we calculated a value for $(\delta_1 - \delta_2)^2$ of $5.95\, cal/cm^3$, while the expected theoretical value should be $0.59\, cal/cm^3$ for the PFO/i-OCT system. This discrepancy is in agreement with the general known behavior of fluorocarbon/hydrocarbon mixtures, that still remains unexplained. By comparing our data for the PFO/i-OCT system to the results published for similar mixtures, such as perfluoro-n-heptane in various n-alkanes or perfluoro- methylcyclohexane in n-octane[76, 77], we find that T_c decreases in the case of iso-octane. This lowering has to be related to the presence of five methyl end groups in the iso-octane molecule - instead of the two terminal CH_3 residues as in n-alkanes - which increases the number of "contact points" with the fluorinated molecule, as described in the "segment treatment"[78, 79], and the interactions of the two chemicals as well.

In Fig.3-2 we reported the phase diagrams of $PFO/i-OCT/F_8H_{16}$ systems, plotting the phase separation temperature T as a function of the copolymer volume fraction ϕ_{COP}. For each curve the volume fraction ratio between PFO and i-OCT(r_v) has been kept constant, as ϕ_{COP} ranges between 0 and 80%. Curve a corresponds to $r_v = 2.8$ and curve b refers to $r_v = 0.81$, which is the exact ratio between the volumes

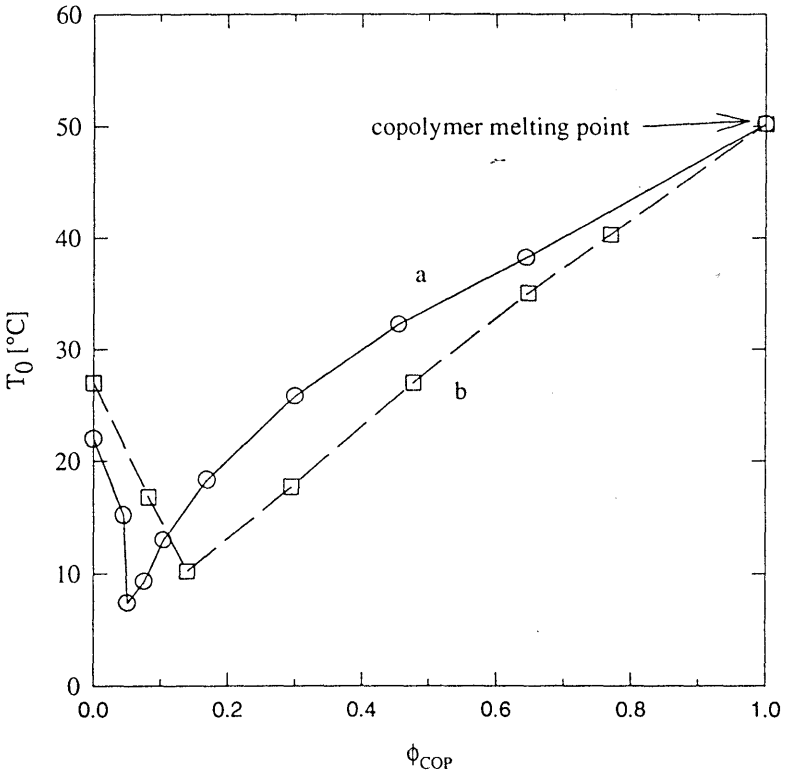

Figure 3-2: Liquid-gel phase transition curves: curve a, $\phi_{PFO}/\phi_{i-OCT} = 2.8$; curve b, $\phi_{PFO}/\phi_{i-OCT} = 0.81$.

of the fluorinated and the hydrogenated blocks in the copolymer[72].

The phase diagram shows a characteristic curve, with a minimum at $\phi_0 \approx 0.05$ for curve a and 0.15 for curve b. This fact, which is quite unusual for low molecular weight compounds, can be explained by invoking some components segregation in the liquid phase, above the phase separation line. Such behavior has already been described for other systems, such as $F_{12}H_{10}$ in n-octane, and confirmed by light-scattering measurements[70]. The presence of this minimum indicates that the $PFO/i - OCT/F_8H_{16}$ system changes its state depending on the amount of copolymer. In fact for $\phi_{COP} < \phi_0$ the curve indicates the value of the liquid-liquid phase separation temperature, T_0, as a function of ϕ_{COP}. The addition of copolymer causes the progressive lowering of the demixing temperature, and therefore enhances the mutual solubility of PFO and i-OCT. For $\phi_{COP} > \phi_0$ instead of the liquid-liquid separation we observe a liquid-gel phase transition at T_g. At a given ϕ_{COP} value, for $T > T_g$ we have a homogeneous and clear solution that transforms into a white solid gel when the sample is cooled down below T_g. This gel is constituted by copolymer ribbons entrapping the two solvents, and extended in a tridimensional network. The addition of copolymer has the effect to increase the lateral size of the ribbon, and arises the temperature required to "melt" the lamellar structure. Moreover the evidence that curve "b" lays below curve "a" for $\phi_{COP} > 10\%$ indicates that when the volume fraction ratio of PFO over i-OCT is exactly equal to the ratio between the molecular volumes of the two blocks in the copolymer(F_8/H_{16}), the most favorable structure is formed as far as the mutual packing hindrances are concerned. Therefore microdomains formed when $r_v = 0.81$ are the most stable and most closely packed, at least with respect to the aggregates produced at $r_v = 2.8$.

Fig.3-3 reports the static intensity I of light scattered at $\theta = 90°$ by $PFO/i - OCT/F_8H_{16}$ mixtures at $22°C$ as a function of the volume fraction ratio r_v for different values of X; curves a, b and c correspond to $X = 20$, 10 and 5 respectively. Each plot shows a peak corresponding to a value of r_v that is comparable to the critical value obtained from phase diagrams, as reported in Table 3.1. The maximum intensity decreases in a significant way when the copolymer is added to the mixture. The angle-

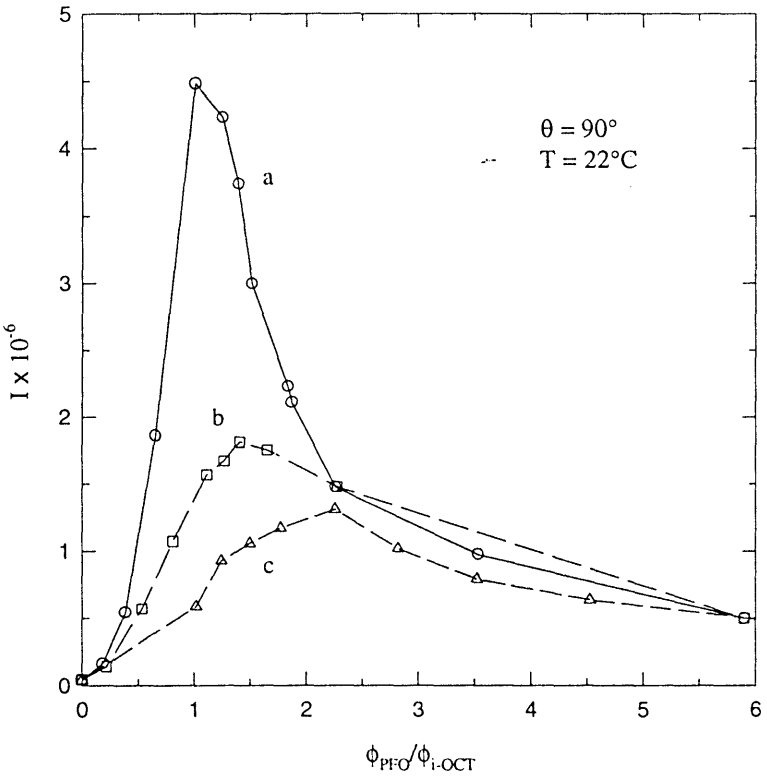

Figure 3-3: Static intensity versus ϕ_{PFO}/ϕ_{i-OCT} at $\theta = 90°$ and 22°C: curve a, $X = 20$; curve b, $X = 10$; curve c, $X = 5$.

dependence of I(dissymmetry coefficient) is almost negligible, which is probably due to the small size of these particles, rather than to a nearly spherical geometry. The presence of the intensity peak has to be ascribed to the critical scattering arising from phase separation of the PFO/i-OCT mixture. Considering that the addition of the copolymer enhances the mutual solubility of the two pure solvents, it is evident that the higher the amount of copolymer in the mixture, the lower the scattered intensity peak in the plot.

Fig,3-4(a) reports the scattered intensity at 90° measured at 27°C for the $r_v = 2.8$ system as a function of the volume fraction of the copolymer. The sharp break in the plot is typical for aggregate formation, and indicates the presence of scattering particles for $\phi_{COP} > 3\%$, which can be assumed as the minimum amount of F_8H_{16} required to form aggregates in the mixed solvent at this temperature. This data is in good agreement with the value indicated in the phase diagram (see Fig.3-2) for which the gel-liquid transition curve reaches its minimum.

Dynamic light-scattering measurements on the liquid samples indicate the presence of weakly interacting particles. The hydrodynamic radius of the equivalent sphere R can be calculated according to the following expression:

$$D = \frac{k_B T}{6\pi\eta_0 R} \tag{3.3}$$

where k_B, T, D and η_0 are the Boltzmann constant, the temperature, the translational diffusion coefficient and the viscosity of the solvent respectively[72]. By substituting the correct values in Eq.3.3, the average hydrodynamic diameter ranges around 30Å(see Fig.3-4(b)). In the case of samples with $r_v = 0.81$ at 27°C, the static intensity steeply increases when the copolymer volume fraction decreases, before the occurring of the complete phase separation of the two solvents. This relates to the critical scattering due to phase separation, and this effect hides any possible scattering deriving from aggregates formation.

Birefringence has been detected in all $PFO/i-OCT/F_8H_{16}$ gel samples observed between crossed nicols(Fig.3-5). The solid gel shows the maltese cross pattern that

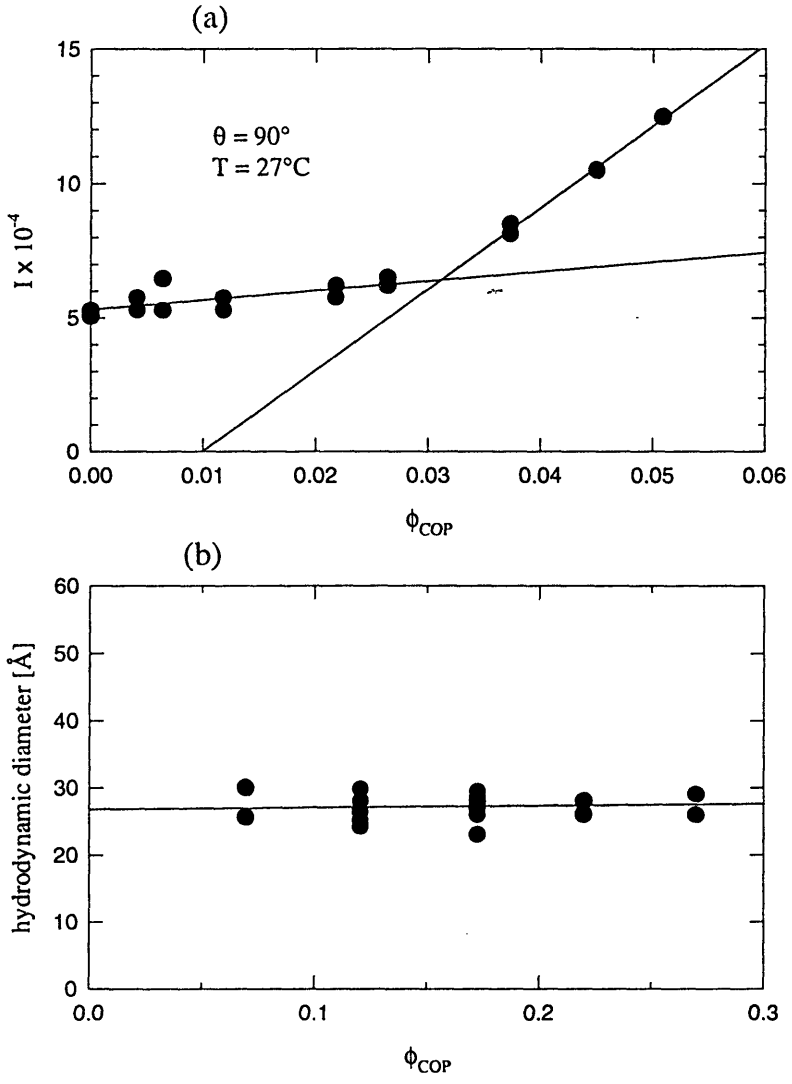

Figure 3-4: (a) Static intensity at $\theta = 90°$ and 27°C versus copolymer volume fraction for $\phi_{PFO}/\phi_{i-OCT} = 2.8$. (b) Hydrodynamic diameter of scattering particles as a function of copolymer volume fraction for $\phi_{PFO}/\phi_{i-OCT} = 2.8$.

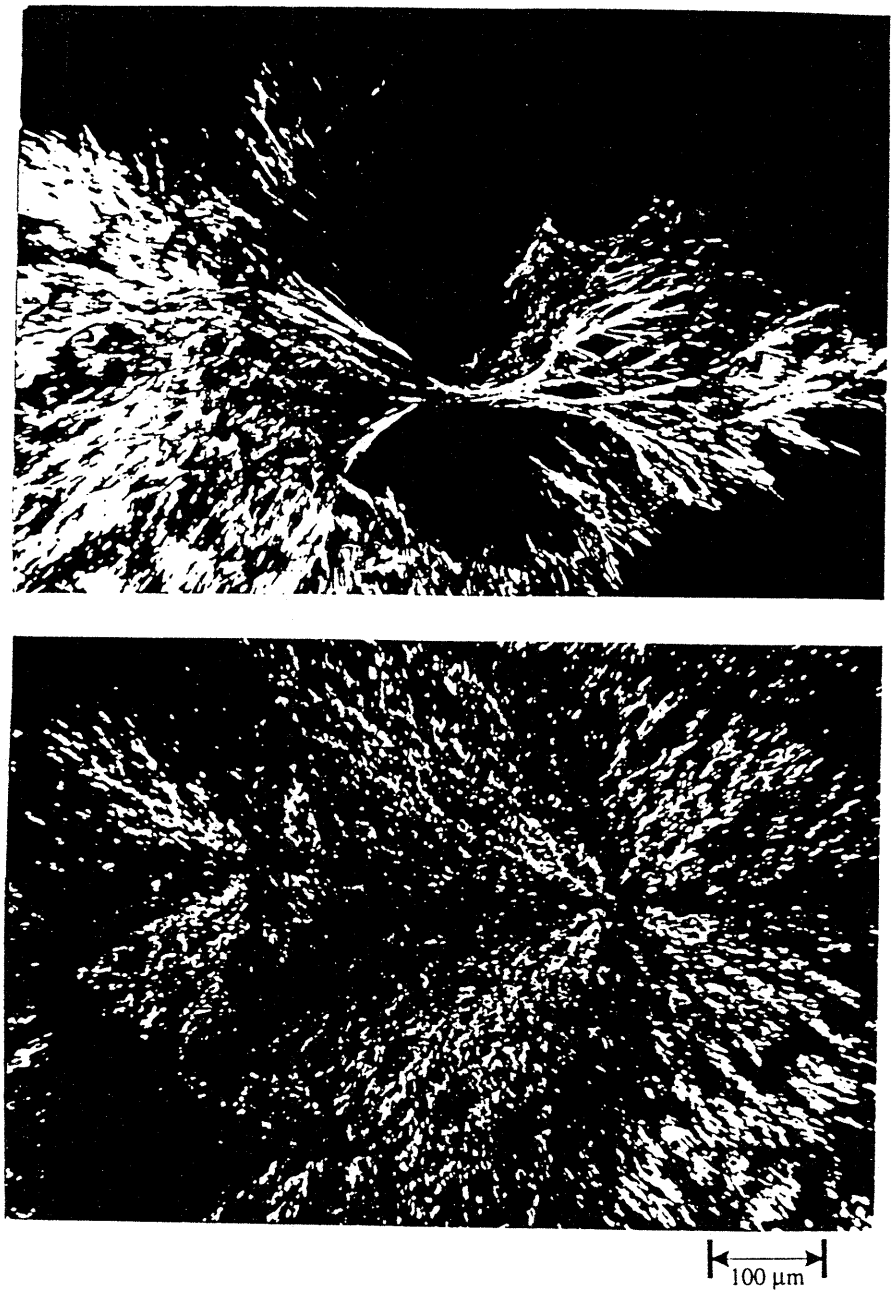

Figure 3-5: Birefringent structures indicating the presence of spherulites in a $PFO/i-OCT/F_8H_{16}$ mixture(gel phase).

is typical of spherulites, and also in the liquid phase the sample rotates the plane of polarized light, indicating the presence of compartimentalized microdomains. Similar observations on semifluorinated n-alkanes have been made and published by other authors[62]. This behavior has been reported as the effect of preferential orientation of microstructures in radial symmetry.

3.4 Model of the liquid

In a Small-Angle X-Ray-Scattering experiment, the absolute scattering intensity can be expressed as:

$$I(Q) = n_p V_p^2 \tilde{P}(Q) \bar{S}(Q) \tag{3.4}$$

where n_p is the particle number density, V_p is the volume of the particle, $\tilde{P}(Q)$ is the particle structure factor and $\bar{S}(Q)$ is the inter-particle structure factor.

For a 3-layer cylindrical particle with two equal-thickness outer layers, the orientational averaged particle structure factor can be written as:

$$\begin{aligned} \tilde{P}(Q) &= \int_0^1 \left\{ \frac{(\rho_{out} - \rho_{sol}) \sin\left[\frac{1}{2}Q\mu(H+2F)\right] - (\rho_{out} - \rho_{in}) \sin\left(\frac{1}{2}Q\mu L\right)}{\frac{1}{2}Q\mu(H+2F)} \right\}^2 \\ &\quad \times \left[\frac{2J_1(QR(1-\mu^2)^{1/2}}{QR(1-\mu^2)^{1/2}} \right]^2 d\mu \end{aligned} \tag{3.5}$$

where H is the thickness of the middle layer, F is the thickness of the outer layer, R is the cross section radius of the cylinder and ρ_{out}, ρ_{in} and ρ_{sol} are the scattering length density of the outer layer, of the inner layer and of the solvent respectively. Because of the free rotational motion of the cylinders in the liquid phase, we assume that these particles interact with each other through an equivalent hard sphere potential[80] with an apparent hard sphere diameter a and a corresponding volume fraction η of the hard spheres in the solvent. This results in an inter- particle structure factor $S(Q, a, \eta)$ used in Eq.3.4.

According to the phase diagram study, light-scattering and birefrnigence experiments, we were able to elaborate a model for the formation of the gel phase from the liquid mixture. In the liquid phase, for $\phi_{COP} > 3\%$, copolymer molecules arrange in micellar aggregates, exposing the fluorocarbon units to the external solvent, while the hydrocarbon chains remain confined in the micellar core. In this way the interactions between fluorocarbon and hydrocarbon blocks are minimized.

As the temperature decreases, approaching the liquid-gel transition temperature(T_g), some of the copolymer molecules aggregate and form the basic micelle unit illustrated in Fig.3-6. In this structure, all fluorocarbon blocks are in a side-by-side arrangement and all hydrocarbon chains are interdigitated. The fluorocarbon-fluorocarbon and hydrocarbon-hydrocarbon interactions are in this way optimized, and the thickness of the micelle is about 44.7$\AA$. Further temperature decrement results in the formation of long, ribbon-like, lamellae that originate from a central "seed" or the basic micelle unit, and extend in many directions(see Fig.3-7) entrapping the solvent in the free cavities and the ribbon-like lamellae retain the thickness of the basic micelle as the lamellar repeating distance. Fig.3-15 shows a top view of the lamellar ribbon, and Fig.3-8 reports the side view, where many layers overlap in a head-by-head arrangement.

In order to explain the experimental results, we propose the following model for micelles. Hydrocarbon tails of F_8H_{16} copolymers aggregate in a side-by-side and tail-to-tail arrangement to form cylinders with a cross section radius of R and a length $H + 2F$ as shown in Fig.3-6. H results to be the fully stretched length of the hydrocarbon part of the copolymer, and F is the length of the fluorocarbon segment in F_8H_{16}. The particle volume V_p used in Eq.3.4 can be replaced by:

$$V_p = \pi R^2 (H + 2F) \tag{3.6}$$

The volume fraction η in the $S(Q, a, \eta)$ factor can be related to the particle number density by:

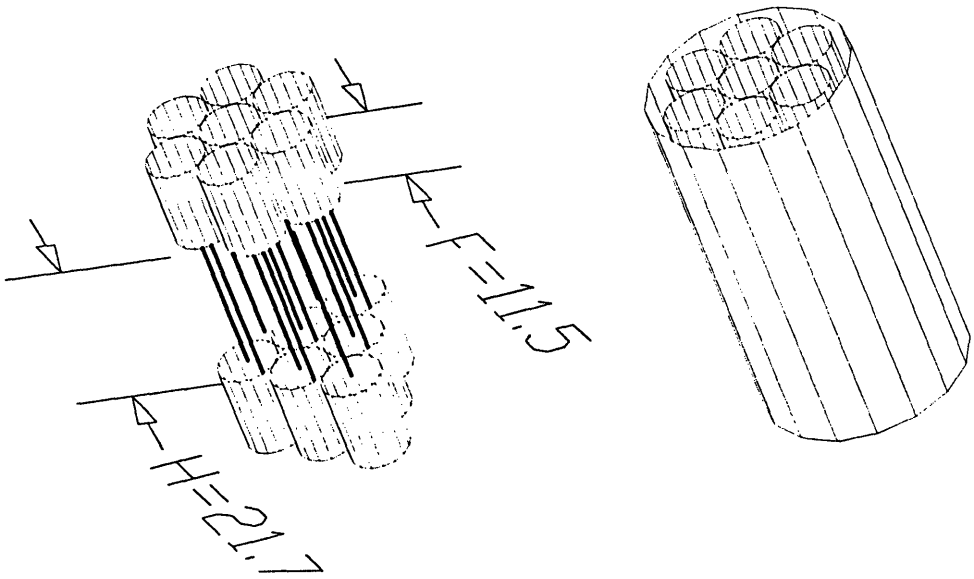

Figure 3-6: Model of the basic micelle unit with the fluorinated blocks(cylinders) closely packed side-by-side and the hydrocarbon chains(thick lines) interdigitated in the internal region.

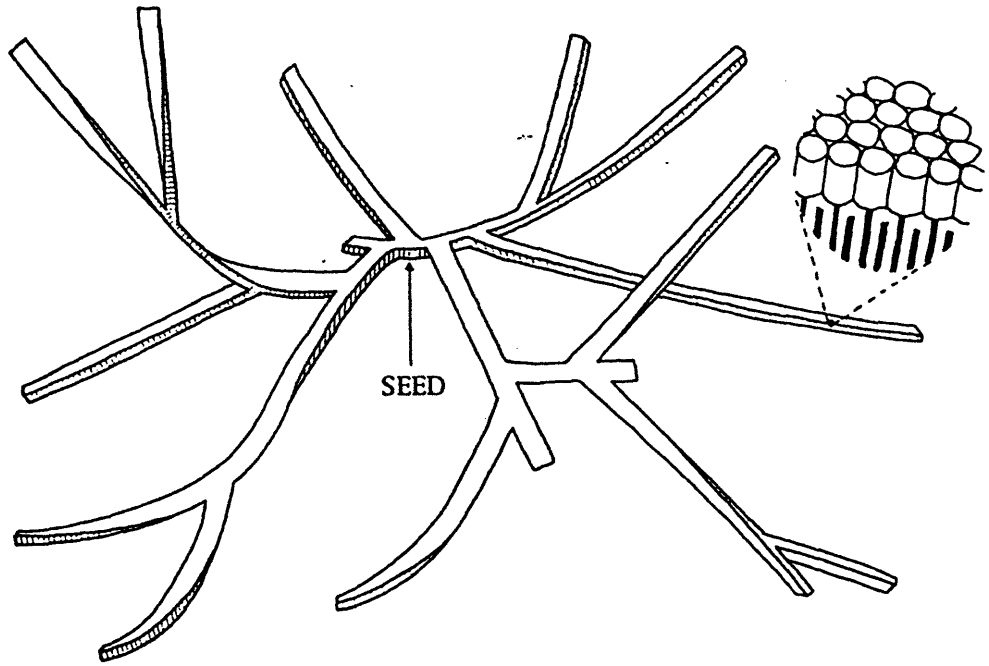

Figure 3-7: Ribbon-like schematic structure for the aggregation of copolymer molecules in the gel phase. White regions indicate top surfaces: shaded regions represent side surfaces. The structure originates from a "seed" and extends randomly into 3-D space, entrapping the mixed solvent between the ribbons.

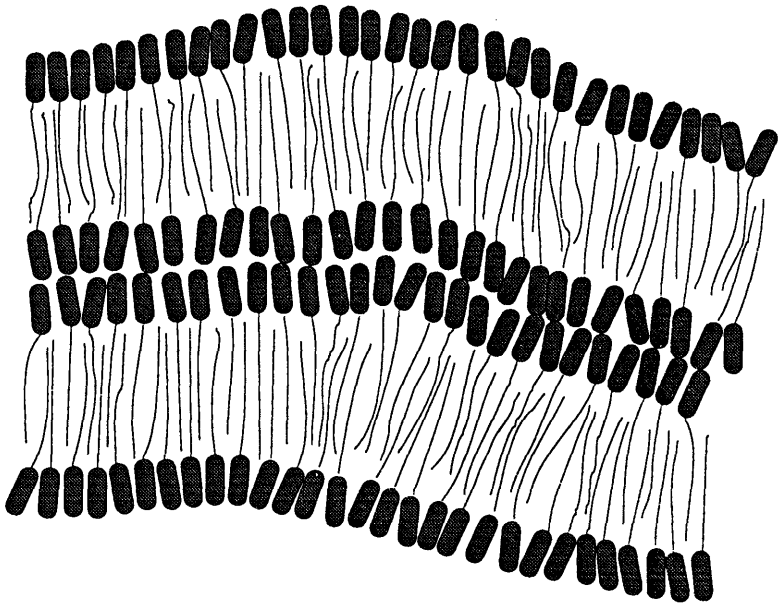

Figure 3-8: Side view of the ribbon-like structure showing the side-by-side and the head-to-head close packing of fluorinated segments. Fluorinated blocks are represented by gray ovals: hydrocarbon chains are indicated by black lines.

$$n_p = \frac{6\eta}{\pi a^3} \tag{3.7}$$

If we rewrite Eq.3.4 using Eqs.3.6 and 3.7, we have:

$$I(Q) = \frac{6\eta}{\pi a^3}[\pi R^2(H+2F)]^2 \tilde{P}(Q)\tilde{S}(Q,a,\eta) \tag{3.8}$$

The solvent in the liquid phase is a mixture of PFO, i-OCT and the rest of copolymer that did not aggregate to form micelles; therefore the scattering length density of the solvent can be written as:

$$\begin{aligned} \rho_{sol} &= \frac{\rho_{i-OCT}}{1-\phi_m}\left[\frac{V_{HC}}{V_{FC}+V_{HC}}(\phi-\phi_m)+\frac{1-\phi}{r_v+1}\right] \\ &+\frac{\rho_{PFO}}{1-\phi_m}\left[\frac{V_{FC}}{V_{FC}+V_{HC}}(\phi-\phi_m)+\frac{r_v(1-\phi)}{r_v+1}\right] \end{aligned} \tag{3.9}$$

where ρ_{i-OCT} is the scattering length density of iso-octane, and ρ_{PFO} is the scattering length density of perfluorooctane, V_{HC} is the volume of the hydrocarbon segment of the copolymer, while V_{FC} is the volume of the fluorocarbon block of the F_8H_{16} molecule, ϕ is the actual volume fraction of the copolymer, $\phi_m = n_p V_p$ is the volume fraction of micelles in the solution. For the SAXS data analysis the variables in the fitting routine were the radius R, the hard sphere diameter a, and the hard sphere volume fraction η.

3.5 Models of the Gel

The common understanding of the structure of a gel indicates that there are cross-linked long molecules, which constitute a entangled network with cavities. The liquids or solvents fill those cavities to complete the formation of a gel. The cross-linking is often a result of chemical bonding which gives rise to a fairly strong network structure in an ordinary gel. This chemical bonding can be hydrogen bonding, solvation, etc..

In our gel samples, there are long fibers grown by successive crystallization of

F_8H_{16} copolymers. These long fibers play the role of the long molecules in an ordinary gel so as to form the porous networks entrapping the liquids among them. The interaction between fibers is considered to be physical contact and the subsequent cross-crystallization or cross-aggregation. Thus the strength of our gel depends on the density of the fibers, since the amount of contacts between fibers increases with the density of fibers. The entrapped liquids among the long fibers then contain the rest of the materials in the system which are not involved in the crystallization processes: PFO, i-OCT, free F_8H_{16} monomers, and F_8H_{16} micelles formed by the free monomers in the liquids. For the SAXS data analysis, a realistic mathematical model should include all the scattering objects in the samples. Therefore, there are two parts in the model we propose for our gel samples, the first part is the crystal part which comes from the scattering of the F_8H_{16} crystalline networks in the gel, and the second part is the liquid part which represents the scattering of micelles in the entrapped liquids.

The scattering patterns we obtained in the experiments are the results of the scattering-length-density contrasts between different domains of the materials. Beside the contrasts inside the crystals and micelles due to the fine structures, there are contrasts between either the crystals or the micelles and the solvents. So it is important to define the solvents in our sample systems. To the crystalline fibers, the solvents are simply the liquids of homogeneous mixture of four components as described above. And to the micelles, the solvents are the homogeneous mixture of three components: PFO, i-OCT, and F_8H_{16} monomers.

According to Fig.3-1, for $X = 10$, $T_c \approx 10°C$. Which means at temperature above $10°C$,for any value of x_1, the mixture is a single phase liquid or the mixture is homogeneous. For the current samples under investigation, there are two sets of data. One set has data with the volume ratio of PFO to i-OCT $r_v = 2.8$ at $10°C$ but different F_8H_{16} volume fraction ϕ_{COP}, and another set has data with $r_v = 0.81$ and $\phi_{COP} = 0.6$ but at different temperatures above $10°C$. For $r_v = 2.8$ and $X = 10$, ϕ_{COP} can be calculated to be 7.2%, meaning the mixture will be a one phase liquid with 7.2% of F_8H_{16} in the mixture. And the increase of ϕ_{COP} will decrease X so as to decrease T_c and make the mixture further into the one phase region. For $r_v = 0.81$

and $X = 10$, $\phi_{COP} = 14.3\%$. Since we know F_8H_{16} form micelles in the single liquid phase, the hypothesis of homogeneous mixture of four or three components can stand if the volume fractions of F_8H_{16} exceed 7.2% and 14.3% for $r_v = 2.8$ and $r_v = 0.81$ data sets respectively.

3.5.1 Crystal Part

In Ref.[70], Höpken proposed a cylindrical model for the crystallization of F_mH_n from both melt and solution based on freeze fracture TEM pictures of $F_{12}H_{20}$ crystals and the fact that the cross-sections of fluorocarbon and hydrocarbon segments are different. This model has not been challenged until in Ref.[60], where we suggested a ribbon model for the networks of F_8H_{16} crystals, which hold the solvents and form a gel phase, to explain the birefringence pictures of the gel.

In this section, we will write down the mathematical formula for both of the models and we will compare the results of data analysis using these two different models in next section.

In a small-angle X-ray-scattering experiment, the absolute scattering intensity of a system of independent particles can be expressed as

$$I(Q) = n_p \tilde{P}(Q) \tag{3.10}$$

where n_p is the particle number density, Q is the magnitude of the scattering wave vector $\vec{Q}$, and $\tilde{P}(Q)$ is the orientationally-averaged particle structure factor. The orientationally-averaged particle structure factor can be expressed as

$$\tilde{P}(Q) = \frac{V_p^2}{4\pi} \int_{-1}^{1} d\mu \int_0^{2\pi} d\phi |A(\vec{Q})|^2 \tag{3.11}$$

in spherical coordinates$(r, \theta, \phi, \mu = cos\theta)$, where V_p is the volume of the particle and

$$A(\vec{Q}) = \frac{1}{V_p} \int_{V_p} d\vec{r} (\rho(\vec{r}) - \rho_s) e^{i\vec{Q}\cdot\vec{r}} \tag{3.12}$$

The integration above is a volume integral over one particle and $\rho(\vec{r})$ is the scattering length density of the particle at location $\vec{r}$, and ρ_s is the scattering length density of the solvents.

For both cylindrical and ribbon models, the first step in constructing the mathematical formula is to define the particle for the integration in Eq.3.12. Therefore, we define the particle by chopping the long fibers, either with a cylindrical cross-section or with a rectangular cross-section, into smaller building bricks with straight edges. Fig.3-9 is a two-dimensional schematic representation of the idea of chopping a branched long fiber into small bricks, and each brick has a rectangular shape which facilitates the mathematical modeling.

In the cylindrical model, it is unlikely that the cylindrical fibers can have branches, as a result, there are many very long and coiled cylindrical fibers tangled together to form the networks for the gel. Fig.3-10(a) shows one long cylindrical fibers and the details of a cylindrical brick. Fig.3-10(b) shows a small section of the branched ribbon structure and the details of a rectangular brick resulting from the chopping of the structure. The repeating distance of the layering structure in both of the models was calculated independently by examining the position of the Bragg peaks in the SAXS scattering data. The result is $\sim 45\mathring{A}$, which equals twice the length of the fluorocarbon segment of the copolymer plus the length of the hydrocarbon segment of the copolymer. As a result, in both of the models, the internal structures are fixed with a bilayer arrangement of the copolymers, with fluorocarbon segments closely packed side-by-side and head-by-head and hydrocarbon segments interdigitally stuffed in between the fluorocarbon layers.

Cylindrical Model

For a cylindrical shape brick with internal structure as described in Fig.3-10(a),

$$A(\vec{Q})_{cyl} = \frac{1}{V_{pcyl}} \int_{-L/2}^{L/2} dz\ e^{iQ_z z} \int_0^{2\pi} d\theta \int_0^R dr\ r[\rho(r) - \rho_s] e^{iQ_\perp r cos(\theta)}$$

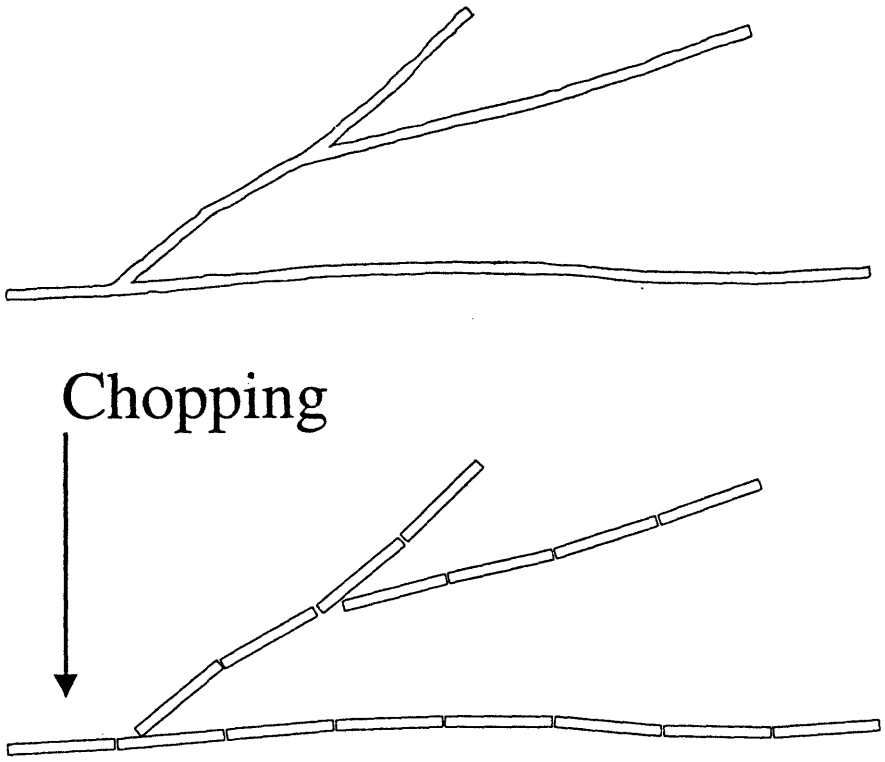

Figure 3-9: Schematic showing a branched fiber and the results of chopping the fibers into small bricks with straight edges.

$$= \frac{sin(Q_z L/2)}{Q_z L/2} \left\{ (\rho_f - \rho_h) \left[\sum_{j=1}^{l} \left(\frac{jt}{R}\right)^2 \left(2\frac{J_1(Q_\perp jt)}{Q_\perp jt}\right) \right.\right. \tag{3.13}$$
$$\left. - \sum_{j=0}^{l} \left(\frac{jt+H}{R}\right)^2 \left(2\frac{J_1(Q_\perp(jt+H))}{Q_\perp(jt+H)}\right) \right]$$
$$\left. + (\rho_f - \rho_s) \left(2\frac{J_1(Q_\perp R)}{Q_\perp R}\right) \right\}$$

where $V_{pcyl} = \pi R^2 L$ is the volume of the cylindrical brick, $t = 2F + H$ is the thickness of one bilayer, F and H are the lengths of fluorocarbon and hydrocarbon segments of a F_8H_{16} molecule respectively, L is the length of the cylinder, $R = (l+1)t - F$ is the cross section radius of the cylinder, l is the number of bilayers in the radial direction, and ρ_f and ρ_h are scattering length densities of fluorocarbon and hydrocarbon respectively. $Q_z = Q\mu$ is the component of $\vec{Q}$ in the axial direction and $Q_\perp = Q\sqrt{1-\mu^2}$ is the component of $\vec{Q}$ in the cross section plane.

In spite of the fact that the bricks are connected to each other to form the long fibers and are not allowed to rotate freely, by assuming randomly distributed fibers in 3-dimensional space, the resulting building bricks are indeed uniformly distributed in all the directions with respect to Q. Thus we can use Eq.3.11 to calculate the rotation-averaged particle structure factor. We apply Eq.3.13 to Eq.3.11 and calculate the rotation-average particle structure factor by doing numerical integration. Before we can use Eq.3.10 to calculate the absolute scattering intensity, we need to find out the particle number density for this model, which can be done by using the equation $n_{pcyl} = \frac{\eta_{cyl}}{V_{pcyl}}$, where η_{cyl} is the volume fraction of cylindrical bricks in the sample. The adjustable variables in this model include L, l, η_{cyl}, and ρ_s.

Ribbon Model

For a rectangular brick with internal structure as shown in Fig. 3-10(b), with length L, width W and thickness nt, where n is the number of bilayers. The volume of a single brick $V_{prib} = n \times L \times W \times t$, $n_{prib} = \frac{\eta_{rib}}{V_{prib}}$ where η_{rib} is the volume fraction of the rectangular bricks, and

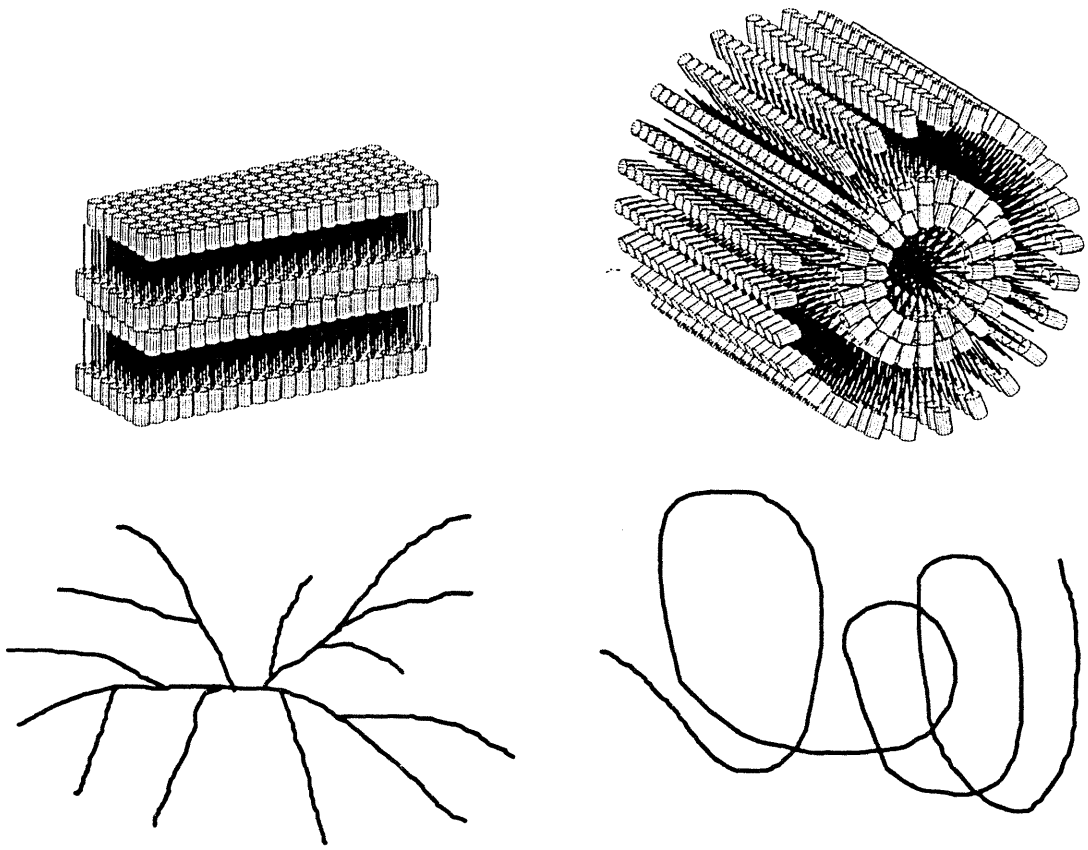

Figure 3-10: Schematic showing one piece of the random ribbons, the chopping of the ribbon, and the details of a building brick.

$$
\begin{aligned}
A(\vec{Q}) &= \frac{1}{nLWt}\int_{-L/2}^{L/2} dx\, e^{iQ_x x}\int_{-W/2}^{W/2} dy\, e^{iQ_y y}\int_{-nt/2}^{nt/2} dz\, [\rho(z)-\rho_s]e^{iQ_z z} \\
&= \frac{1}{nt}\,\frac{sin(Q_xL/2)}{Q_xL/2}\,\frac{sin(Q_yW/2)}{Q_yW/2} \\
&\quad\times\left\{2F(\rho_f-\rho_s)\left\{cos\left(Q_z\frac{nt-F}{2}\right)\frac{sin(Q_zF/2)}{Q_zF/2}\right.\right. \\
&\quad\left.+2\left[\frac{n-1}{2}-m+\sum_{j=1}^{m}cos\left(Q_z\frac{n-2j}{2}t\right)\right]\frac{sin(Q_zF)}{Q_zF}\right\} \\
&\quad\left.+2H(\rho_h-\rho_s)\left[\frac{n}{2}-m+\sum_{j=1}^{m}cos\left(Q_z\frac{n+1-2j}{2}t\right)\right]\frac{sin(Q_zH/2)}{Q_zH/2}\right\}
\end{aligned}
\tag{3.14}
$$

where $m = \frac{n}{2}$ for even n, $m = \frac{n-1}{2}$ for odd n. $Q_x = Q\sqrt{1-\mu^2}cos\phi$ is the component of $\vec{Q}$ along the length of the brick, $Q_y = Q\sqrt{1-\mu^2}sin\phi$ is the component of $\vec{Q}$ along the width of the brick, and $Q_z = Q\mu$ is the component of $\vec{Q}$ in the direction perpendicular to the layers. The calculation of absolute scattering intensity can be carried out following the same steps as in the cylindrical model. L, W, n, η_{rib}, and ρ_s are the adjustable variables in this model.

3.5.2 Liquid Part

The liquids in our sample are homogeneous mixtures of four components as mentioned above and the micellar model described in the previous section is used to analyze both liquid and gel phases data. Those micelles in the liquid serve as the seeds for the growth of long and branched ribbon-like crystalline fibers as we decrease the temperature below the liquid-gel transition temperature. It combines with the ribbon model to give an integral description of the system in both liquid and gel phase.

The micellar model gives satisfactory results compared to the scattering data for low concentration of micelles in the liquids but it fails at high concentration of micelles, due to the more complicated interaction among the micelles where a simple two-body spherical model such as the sticky-hard-sphere model does not apply anymore. Therefore, we can use this model to analyze the low concentration liquid phase data

or combine the model with ribbon model to analyze the low concentration gel phase data. This model is also applicable to the high concentration gel phase data if the micelle concentrations in the entrapped liquids are sufficiently low. Those data sets with $\phi_{COP} = 0.6$ at 20 and $30°C$ fall into that category since at those temperatures, the sample is far into the gel phase, and most of the copolymers are included in the fiber networks, there are much less monomers left in the entrapped liquid to form micelles.

For the data set that the micellar model doesn't apply, we use the measured scattering curve, which was taken at higher temperature so that the sample was in the liquid phase, as the empirical curve for the scattering of the micelles. For example, for the sets of data with $r_v = 0.81$ and $\phi_{COP} = 0.6$, at $40°C$, the sample was in its liquid phase, and at $30°C$ the sample became a gel. Therefore, we used the scattering curve at $40°C$, with an adjustment amplitude factor to correct for the volume difference, as the background of the scattering curve at $30°C$, and use the cylindrical or ribbon model to calculate the scattering of the crystalline fibers. This data set is the only one we analyzed with the cylindrical model because the micellar model for the liquid is in fact a counter part of the ribbon model. The counter part of the cylindrical model doesn't exist.

The combined total absolute scattering intensity can be written as

$$I_{total}(Q) = C\left[(1-\eta)I_{liq}(Q) + \frac{\eta}{V_p}\tilde{P}(Q)\right] \tag{3.15}$$

where C is an adjustable constant to account for the possible volume expansion of the sample cells which were under high vacuum during the scattering experiment, and it also accounts for the decrease of the volume of the sample due to the formation of the gel. The empty space in the sample cell due to the formation of the gel always located at the center of the cell where the x-ray beam was going through, since it was the last position of the sample cell where the temperature would reach the set point. Fraction of empty space at the location of the x-ray beam was not controllable, the variable C in Eq.3.15 includes also this consideration. $I_{liq}(Q)$ is the absolute scattering

Table 3.2: List of variables extracted from the data analyses for different liquid samples.

r_v	ϕ_{COP}	Temp.	R	a	τ	ϕ_m
2.8	0.10	10°C	12.8	38.3	0.15	0.024
	0.16		12.6	38.8	0.24	0.031
	0.26		9.37	33.7	1.80	0.047
	0.38		7.46	35.8	15.9	0.053

intensity of the entrapped liquid, it can be calculated using the micellar model or can be the empirical number obtained at higher temperature where the sample is in its liquid phase. η is the volume fraction of the crystals in a gel, V_p is the volume of a single brick, and $\tilde{P}(Q)$ is the particle structure factor for either cylindrical model or ribbon model.

3.6 Results and Discussions

In this section, the first subsection presents the results of liquid phase data analyses using the micellar model. The second subsection is the comparison of the results of data analyses using cylindrical and ribbon models for a gel sample. The third subsection summarize the data analyses of the gel phase data sets using ribbon model. And from the variables extracted in the data analyses, a mechanism for the formation of the gel is suggested.

3.6.1 Liquid Phase Data Analyses

The model for micelles was used to fit four SAXS data obtained from four samples at 40°C in the liquid phase, with an i-OCT/PFO volume fraction ratio of 2.8, and copolymer volume fraction 10, 16, 26, and 38% respectively. The resulting variables are listed in Table 3.2 and the fitting curves are reported in Fig.3-11.

The results of the calculations are in excellent agreement with the data; the small number of ϕ_m compared to the actual ϕ_{COP} represents that only a small amount of copolymer molecules do aggregate in the liquid phase. From the radius of the

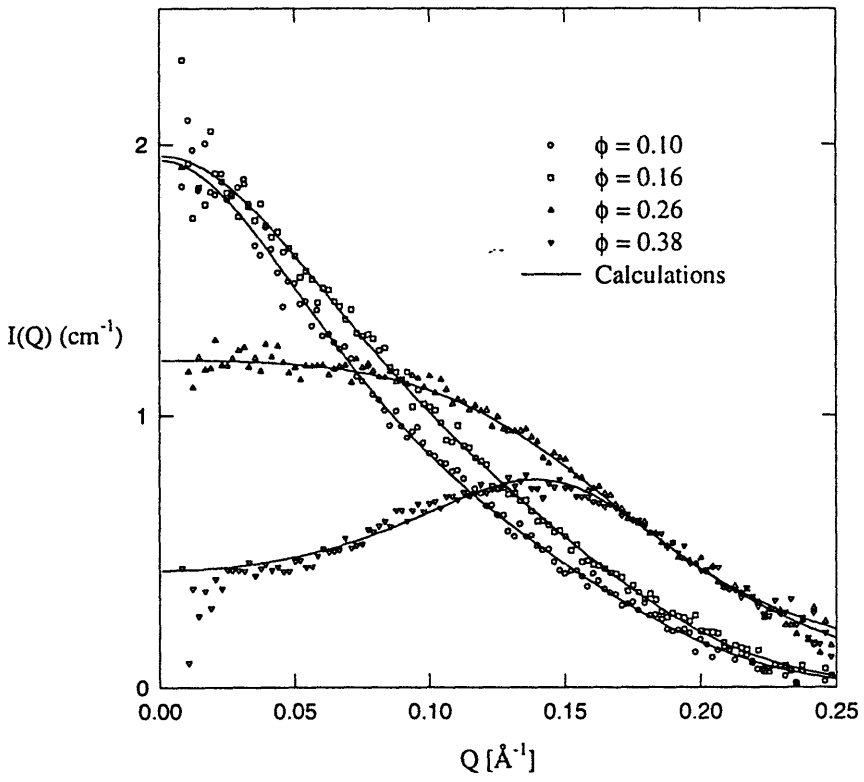

Figure 3-11: SAXS data for four liquid samples at 40°C with $\phi_{cop} = 10\%$, 16%, 26%, and 38%. For each sample the plot shows the experimental absolute intensity $I(Q)$ and the intensity calculated according to the model presented in the text. For these four liquid samples, the parameters resulting from the fits are reported in Table 3.2. The agreement between the experimental intensities and those calculated from the model is excellent.

cylinders, we can calculate the aggregration number to be around 20 and it is larger for the samples closer to T_g. The absolute intensity was calculated taking into account solvent and solute contrast in absolute scale, geometry and volume of particles, the excluded volume effect of the micelle and the peculiar arrangement of fluorocarbon and hydrocarbon blocks in the aggregated structure; the calculated intensity agrees with the experimental values in an excellent way, confirming that the model of the micelle is correct.

3.6.2 Comparison

We used the data set of the sample with $r_v = 0.81$ and $\phi_{COP} = 0.6$ at $30°C$ for the comparison between cylindrical and ribbon models. As described in the previous section, for the data analysis for this particular set of scattering curve, we used the data set of the same sample at $40°C$ as the background scattering which actually came from the scattering of the micelles in the entrapped liquid. The results are displayed in Fig.3-12.

From Fig.3-12(d), The ribbon model gives much better agreement at the small Q region. The discrepancies for both models at Q ranging from 0.05 to 0.12 $\AA^{-1}$ come from the liquid data shown in Fig.3-12(b), which doesn't completely represents the scattering of the micelles for the gel phase data in Fig.3-12(d). For the sharp peak in the experimental data, the ribbon model apparently fits the data better than the cylindrical model. Fig.3-12(a) and (c) display the calculations of the cylindrical model and the ribbon model respectively, the ribbon model produces a major sharp peak which matches the data better than the split peaks produced by the cylindrical model. The ribbon model indicates the brick is $966\AA$ long and the concentration of the ribbons is 5%, which are more appropriate numbers for the approximation of treating the brick as an independent particle in Eq.3.10 than those numbers given by the cylindrical model, $80\AA$ and 15%. Since for the approximation of independent particle to be valid, the interaction between particles needs to be very weak or long-ranged. For the situation in our samples, the bricks are essentially connected to each other, interaction is strong. But if the particles were very long, so that the interaction

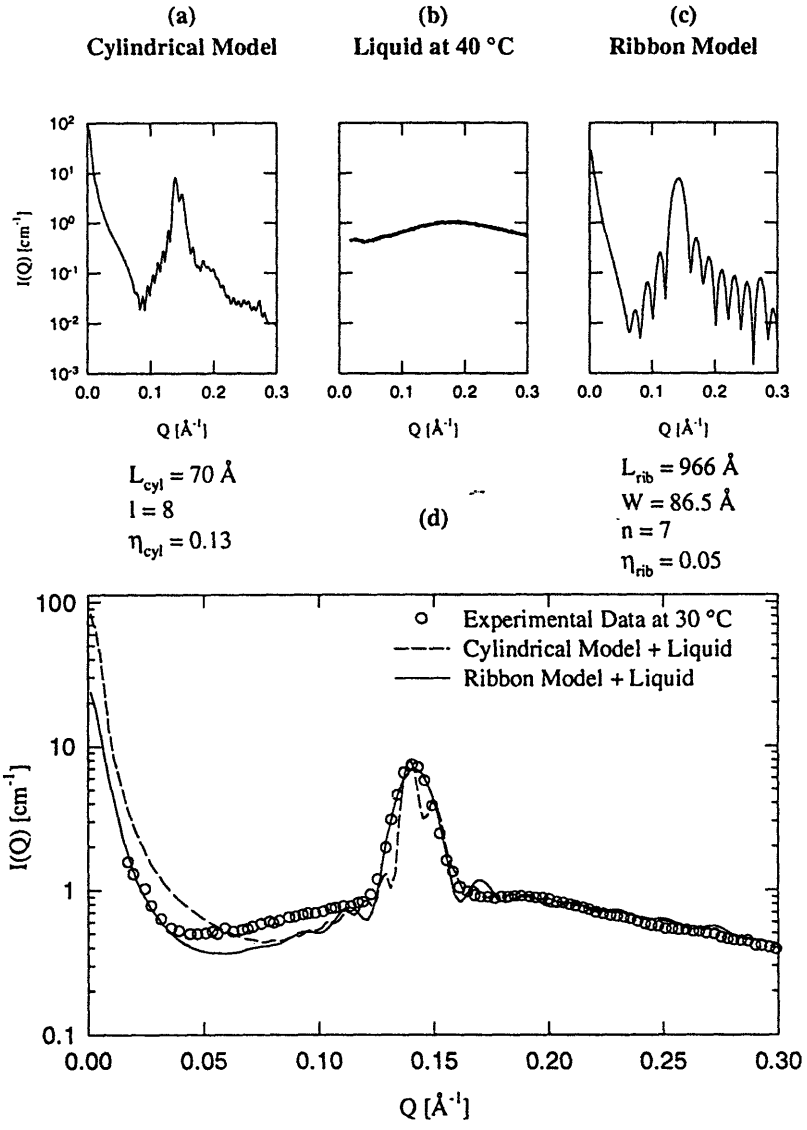

Figure 3-12: The results of the data analyses of using two different crystalline models. (a) is the calculation of the cylindrical model, the numbers blow the plot are the extracted variables for the model; (b) is the liquid phase data for the micelle scattering; (c) is the calculation of the ribbon model; (d) shows the $30°C$ data set of $r_v = 0.81$, $\phi_{COP} = 0.6$, and the resulting curves of combining different models with the micelle scattering of plot (b).

between particles only affects the calculation at very small Q, the approximation is still applicable. For a $966\mathring{A}$ long brick in our case, the effect of inter-particle interaction is mostly on $Q < \frac{2\pi}{966\mathring{A}} = 0.0065\mathring{A}^{-1}$, which is much less than the first Q position in our data.

The comparison above encouraged us to use the ribbon model to continue the data analyses with the rest of our data sets.

3.6.3 Gel Phase Data Analyses

We used the ribbon model combined with the micellar model to analyze the rest of the data sets. Fig.3-13 is an example of the comparison between the experimental data and calculation. It also displays the crystal part and liquid part of the calculation. The crystal part produces the small-Q peak and the diffraction peak around $Q = 0.14$, the liquid part constitutes the smooth portion of the scattering curve. The multilayer structure of the building brick in our ribbon model also produces oscillations at all Q's as depicted in the figure. These oscillations are not significant for the data sets with relatively low F_8H_{16} volume fractions, namely the data sets with F_8H_{16} volume fractions of 0.1, 0.16 , and 0.26 respectively. Fig. 3-14(a) displays the results of the data analyses for the samples of $r_v = 2.8$ and three different F_8H_{16} volume fractions at $10°C$. The agreements between the experimental data and calculations are satisfactory. The oscillations will dominate the scattering curve as the crystals dominate the structure of the gel. Fig. 3-14(b) displays the results of the data analyses for the sample of $r_v = 0.81$ at three different temperatures with F_8H_{16} volume fraction equals 0.1. The sample at $10°C$ has most crystals and least liquids, therefore the oscillations are most severe in its theoretical calculation. Another reason for the worse fitting in Fig. 3-14(b) is the breakdown of the liquid model for high ϕ_{COP} samples as explained in a previous section. One of the possible solutions for the oscillations in the calculation is the incorporation of diffuse boundaries between layers and the polydisersed layer thickness. Another possibility is to have a polydisperse particle system instead of the monodisperse system in use in this article. Either approach will involve a lot more computation in the data analysis process.

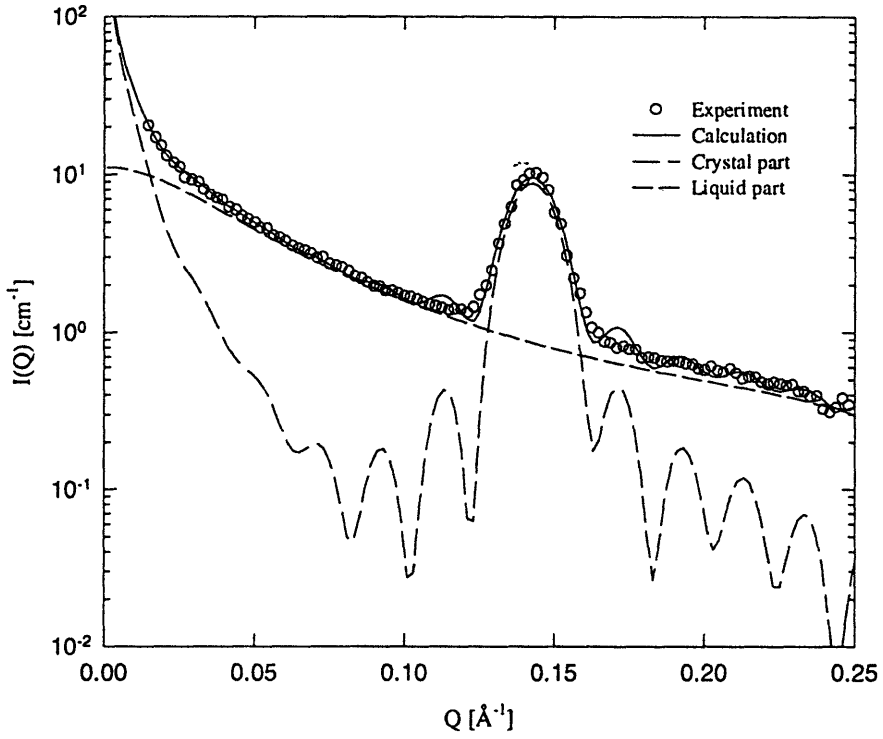

Figure 3-13: The comparison between the experimental data and calculation. Open circles are the experimental data points for the sample with 26% of F_8H16 and r_v = 2.8 at 10°C. Long dashed line is the calculation of the crystal part calculation, short dashed line is the liquid part calculation, and solid line is the combination of them.

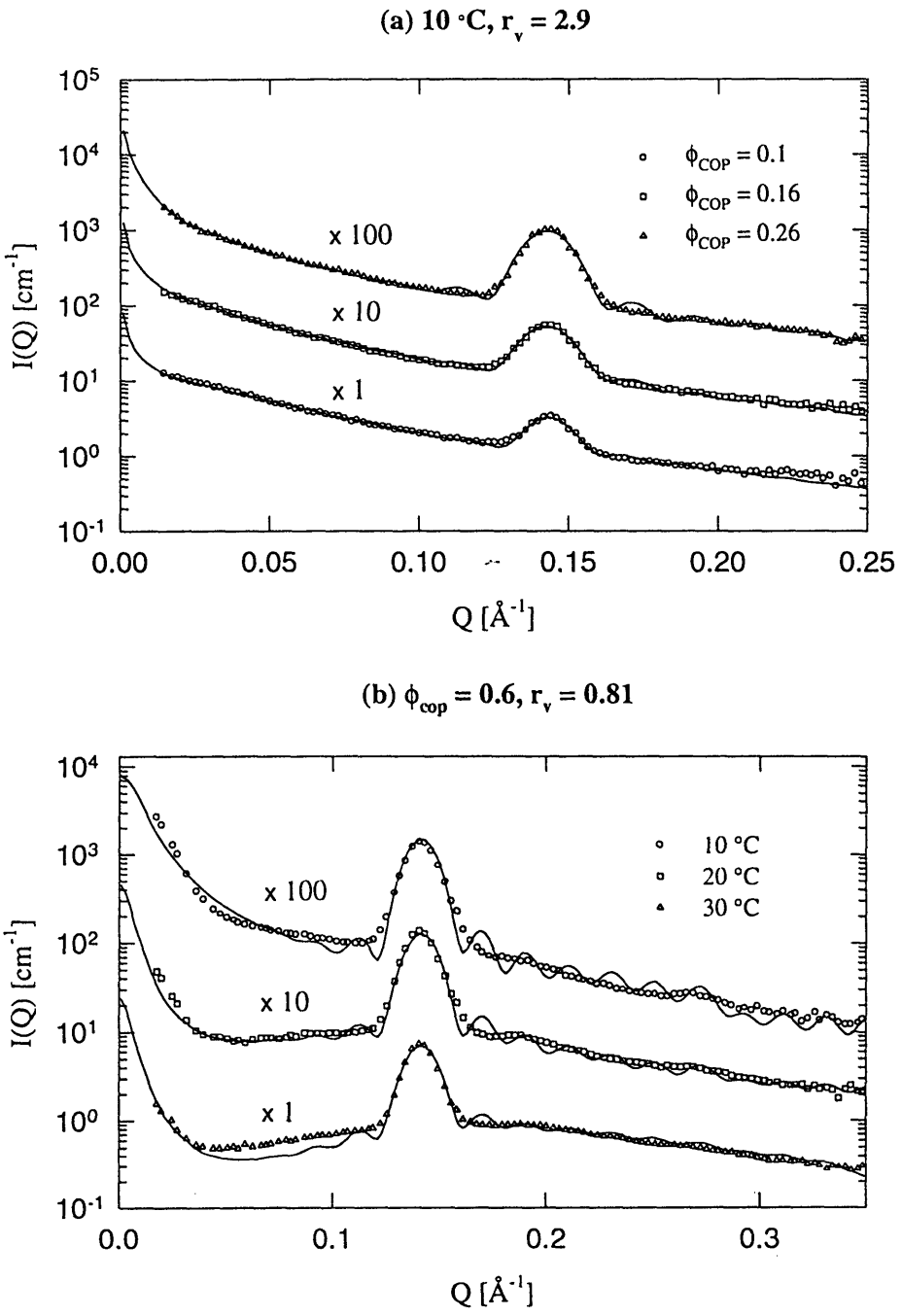

Figure 3-14: The results of data analyses using the combination of ribbon model for the crystal and micelle model for the liquid. Solid line are theoretical calculations. (a) shows results of three lower ϕ_{COP} samples at same temperature. (b) shows results of one high ϕ_{COP} sample at three different temperatures.

Table 3.3: List of variables extracted from the data analyses for different gel samples

r_v	ϕ_{COP}	Temp.	L	W	n	η	ϕ'_{COP}	R	a	τ
2.8	0.1	$10°C$	3150	13.6	8	0.024	0.076	9.08	23.3	0.082
	0.16		2900	41.1	7	0.027	0.133	9.97	27.4	0.106
	0.26		2097	60.6	7	0.063	0.197	9.66	27.3	0.113
0.81	0.6	$10°C$	442	121	7	0.425	0.175	9.85	30.6	0.103
		$20°C$	913	90.1	7	0.107	0.493	7.41	31.8	9.63
		$30°C$	966	86.5	7	0.051	0.549			

ϕ'_{COP} is the volume fraction of F_8H_{16} left in the liquid.

Table 3.3 summaries the variables extracted from the data analyses. It is interesting that the building brick or the unit cell in our crystal model has seven or eight bilayers, which corresponds to 300-350 $\AA$, and it is very long but extremely narrow. The size of the building brick varies with concentration and temperature. It is longer and narrower at low concentration, and becomes shorter and wider at high concentration. On the other hand, it is longer and narrower at higher temperature and shorter and wider at lower temperature. The fibers constructed by longer building bricks are less curly than the fibers constructed by shorter bricks. For the micelle part of the variables, the size of the micelles remains approximately the same and they have similar stickiness regarding the interaction among them. The special case is the data set of $\phi_{COP} = 0.6$, $r_v = 0.81$ at $20°C$. The micellar model is about to breakdown in this case.

3.6.4 Formation of a Gel

As we have mentioned above, the combination of the ribbon model and micellar model for the gel phase is an extension of the micellar model for the liquid phase. They form an integrated description for the whole system, starting at temperature above the liquid-gel transition temperature where the micellar model applies, and ending at temperature below the transition temperature where the gel forms and the combined model applies.

Imagine the liquid phase of the system, the liquid is a homogeneous mixture of four components: perfluorooctane, iso-octane, F_8H_{16} monomers, and cylindrical micelles

formed by F_8H_{16} monomers. As we decrease the temperature in the liquid phase, the radius of the cylindrical micelles will increase and the number of the micelles in the liquid will also increase. When the temperature is lower than the liquid-gel transition temperature, the F_8H_{16} monomers start to attach to the nearest micelles. Due to the particular feature of the micelle, it is easier for the monomers to attach to the micelle from the radial direction than from the axial direction. At the beginning, the micelle starts to grow uniformly in the radial direction, after a while, there are random positions around the peripheral of the micelle that catch more monomers than other positions. These positions form islands around the micelle and deplete the nearest monomers around the micelle at the moment. The tips of these islands are now the closest positions for the monomers surround the micelle to attach to, therefore, these islands grow towards the solvent and become strips. Each strip may split into two strips due to the increasing peripheral area for cylindrical geometry as the radius increases. As the strip grows longer, the monomers can also slowly attach to the strip along the axial direction so as to make the strip thicker. The width of a strip is limited by the competition of neighboring strips. At the end of the formation process, according to the results of the data analyses, these strips become long, curly ribbons with the thickness corresponding to the width of the strips and the width corresponding to the thickness of the strips. Fig.3-15 is a two-dimensional schematic drawing(top view) of a growing structure just described. The final shape of the ribbons depends on the concentration and temperature of the sample as described in the data analyses section. It is reasonable that at lower concentration, there are less seeds to start the growing of a gel, therefore there is less interference on the growing of the ribbons, which makes the ribbons less curly. And due to the fact that there are less monomers in the solvent, the ribbons are thinner.

In the entrapped liquids among these ribbons, there are monomers and micelles. Each micelle is constructed by $\approx$20 monomers. These micelles may be the micelles from the liquid phase which fail to grow or they may be formed during the gel formation process. They keep the mixture of iso-octane and perfluorooctane from phase separation. Table 3.3 lists the volume fractions of ribbons in the samples and the re-

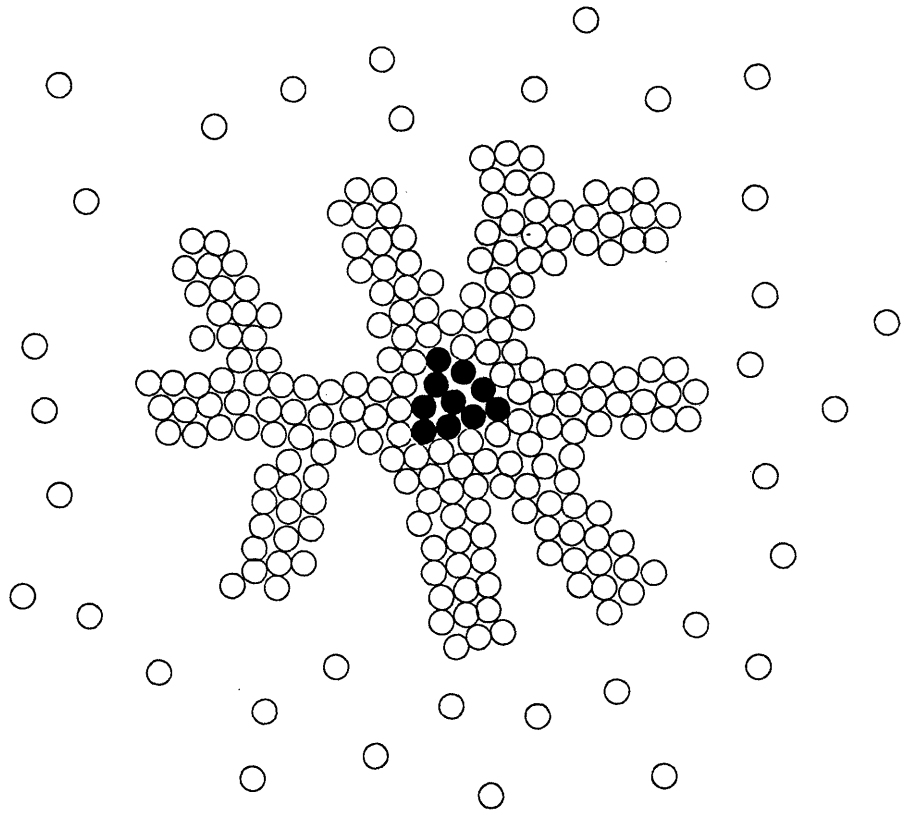

Figure 3-15: A two-dimensional schematic of the growing of the crystalline matrix(ribbons) of a gel

maining F_8H_{16} in the liquid. The second set of numbers all exceed the corresponding thresholds of having homogeneous mixtures as mentioned in the models of the gel section. It verifies the entrapped liquids are indeed one-phase liquids.

3.7 Conclusions

This chapter reports on the phase separation behavior of a perfluorooctane/iso-octane mixture (PFO/i-OCT) and on the effect induced by the presence of a semifluorinated di-block copolymer, $F(CF_2)_8(CH_2)_{16}H$(shortly F_8H_{16}).

The PFO/i-OCT system shows a liquid-liquid phase separation of the two components below a certain temperature(T_0). The coexistence curve indicates the presence of an upper consolute point corresponding to a critical molar fraction of PFO $x_c = 0.321$(volume fraction of PFO $\phi_c = 0.422$) and to a critical temperature $T_c = 27.4°C$.

The addition of even small amounts of semifluorinated copolymer produces a significant lowering of T_c and a relevant broadening of the coexistence curve. Light-scattering measurements do not indicate the presence of stable aggregates in this low-copolymer regime, but only a critical scattering peak, depending on the composition of the mixture. Further addition of copolymer to the liquid mixture results in the formation of a gel phase below a gel-liquid phase separation temperature(T_g).

Phase diagrams of the $PFO/i-OCT/F_8H_{16}$ mixture show that the liquid-gel transition temperature depends on the volume fraction ratio ϕ_{PFO}/ϕ_{i-OCT}, and on the copolymer volume fraction ϕ_{COP}. For ϕ_{COP} smaller than a critical value ϕ_0 the system still separates in two liquid phases, and the copolymer dissolves in the two components. In this case F_8H_{16} is not able to form stable aggregates and has the mere effect to make the PFO/i-OCT mixture more stable, by reducing the mutual incompatibility of the two solvents. On the other hand, for $\phi_{COP} > \phi_0$ the liquid phase produces a solid gel upon cooling. This region of the phase diagram indicates the presence of copolymer aggregates that avoid the liquid-liquid phase separation and compatibilize perfluorooctane and iso-octane even at relatively low temperatures.

Birefringence experiments show the presence of spherulites in the gels, with the typical maltese cross patterns, indicating the formation of compartimentalized domains in the gel phase. Optical activity is revealed also in the liquid samples, and supports the existence of microdomains in liquid phases as well. Quasi elastic light-scattering (QELS) and small-angle X rays-scattering (SAXS) confirm the presence of aggregates in the liquid samples, with an average hydrodynamic diameter of about $30\mathring{A}$.

In order to explain the results a model for the copolymer aggregation in the PFO/i-OCT system in its liquid phase is proposed. In the liquid phase the semifluorinated copolymer aggregates in the mixed solvent and form small micelles. When the temperature is cooled down below the liquid-gel phase transition temperature, the basic micelle unit grows as an extended, ribbon-like, lamella with the fluorinated segments closely packed in a side-by-side and head-by-head arrangement, and the hydrogenated blocks interdigitated in the internal region. Eventually several lamellar layers are formed and produce the birefringent elongated structures that can be observed at the polarizing microscope.

Two models are investigated: cylindrical model and ribbon model, for the microstructure of the gel phase of F_8H_{16} dissolved in a mixture of perfluorooctane and iso-octane. By analyzing a SAXS absolute intensity curve with both models, we conclude the ribbon model is the better representation for our samples.

Using a combined model of ribbon model for the crystal part and micellar model for the liquid part of the gel, we find the fibers in a gel are ribbons with widths of 300-350 $\mathring{A}$, thicknesses of 13.6-121 $\mathring{A}$, and these ribbons have persistent lengths of 442-3150 $\mathring{A}$, depending on the volume ratio of perfluorooctane to iso-octane, the volume fraction of F_8H_{16}, and temperature. Within the persistent lengths, the ribbons are practically straight. We also verify the entrapped liquids among the fibers are one phase liquids composed of perfluorooctane, iso-octane, F_8H_{16} monomers, and micelles.

From the structural parameters obtained in the data analyses, a growing mechanism for the formation of a gel starting from the liquid phase of a sample is proposed. This mechanism gives a reasonable explanation for the final shape of the crystalline

fibers in a gel of the system.

The models used in this article are approximations of a very complicated system. These calculations include two-dimensional integration, therefore the data analyses process is rather time consuming. As mentioned in the data analyses section, it is possible to improve the ribbon model by including polydispersed layer thickness, etc.. However, these additional improvements will result in extremely cumbersome computations. Even at the present level of the model, the overall agreement with absolute intensity data is good enough to warrant a serious consideration.

A FORTRAN computer program of data analysis with various models is listed in Appendix C.

Chapter 4

Conclusions

This thesis contains two different but related works that have been done during my study at MIT. They are related in the sense that the small-angle scattering technique has been utilized in both works to yield important information about the systems under study.

In the first work, a theory based on extensions of Baxter's model has been developed for the phase diagram and SANS data analysis of a droplet microemulsion system, specifically the AOT/water/decane microemulsion system. These extensions include:

1. The incorporation of polydispersity of the size of the droplet in the SANS data analysis.

2. Identifying the specific temperature dependence of the stickiness parameter $1/\tau$ from both theoretical study of Baxter's model and systematic SANS experiments.

3. Identifying the percolation threshold as a point where the cluster size diverges and the experimental temperature dependent effective droplet size.

This theory provides explanation successfully for both the critical and percolation phenomena possessed by the AOT/water/decane microemulsion system.

The second work is the study of short chain copolymer $SF_3(CF_2)_7(CH2)_{15}CH_3$ in a mixed solvent of perfluorooctane and iso-octane. In the solvent, the copolymers form micellar aggregates and gels.

The phase diagram has been determined, which identifies the regions of micellar formation and gel formation. From the information provided by light scattering and birefringence picture, models for the microstructures in each of these regions have been investigated and finalized by the SAXS experiments. It is for the first time that SAXS cross section formula is developed for analysis of gel. This analysis accounts for both the internal structure of the basic scattering unit with the length scale $\sim 45\mathring{A}$, and the random orientation of the ribbons with the length scale $\sim 1000\mathring{A}$.

A mechanism for the growth of the gel in the system is then suggested based on the structural information provided by the analyses of SAXS data.

The test of the Baxter's model, as described in this thesis, has explained all the phenomena in this particular droplet phase region of the system. On the other hand, the copolymer system still has lots of unexplored phenomena that need to be explained. For example, the volume contraction phenomena observed in the gel phase, the possible phase transition in the crystalline structure of the gel, and the effect of different solvents, etc.. More experimental and theoretical evidences are necessary to reconfirm the models presented in the thesis, and the models also welcome further modifications. Although the cylindrical model is not the choice in this study, it can be applied to different systems having cylindrical conformation. The model for the gel phase in this thesis has set a ground of a different approach for the small-angle scattering data analysis of other gels, which is a still wide open field.

The following is a list of publications related to my works toward the degree, which includes publications included in this thesis and publications not included in this thesis, which are studies done with Professor Sow-Hsin Chen and other collaborators, and with my co-supervisor, Professor Xiaolin Zhou. One and a half years ago, Professor Zhou started a project of constructing a neutron reflectometor at MITR II research reactor, since I am interested in instrumentation, we worked together on the design of the spectrometer and generated several new concepts of utilizing the neu-

tron beam in a more efficeint way. It is an exciting experience to put a new idea into a experiment. We have successfully finished one of the experiments and the results will be published soon.

1. "Structure and Dynamics of Water-in-oil Microemulsions near the Critical and Percolation Points,"
   C. Y. Ku, S. H. Chen, J. Rouch, and P. Tartaglia,
   invited to "Thermo-Physical Properties", Colorado, USA, June 1994. *Int. J. Thermophys* **16** (1995) 1119.

2. "The MIT Specular and Diffuse Neutron Reflectometer for the Investigation of Surfaces and Interfaces,"
   C. Y. Ku and X. L. Zhou,
   in Neutron Scattering in Materials Science, eds., D. A. Neumann, T. P. Russell, and B. J. Wuensch, *MRS Proc.* **376** (1995).

3. "Structural Study of the Gel Phase of a Semifluorinated Alkane in a Mixed Solvent,"
   C. Y. Ku, P. LoNostro, and S. H. Chen,
   In manuscript. This will be a paper emphasizing on the construction of gel model as described in the thesis.

4. "Aggregation and Counterion Distribution of Polyelectrolyte: PODMA in Water,"
   C. Y. Ku, S. L. Chang, and S. H. Chen,
   In preparation. We have done SANS experiments in order to determine the dimension of the micelle formed by PODMA with cesium counterions, and SAXS experiments of the same samples for the information on the distribution of the cesium counterions dissolved in water to test the theory of ion condensation.

5. "Effect of a Semifluorinated Copolymer on the Phase Separation of a Fluorocarbon/Hydrocarbon Mixture,"

P. LoNostro, C. Y. Ku, S. H. Chen, and J. S. Lin,
J. Phys. Chem. **99** (1995) 10858.

6. "Ion Correlations in a Micellar Solution Studied by Small-Angle Neutron and X-Ray Scattering,"
Y. C. Liu, C. Y. Ku, P. LoNostro, and S. H. Chen,
Phys. Rev. **E51** (1995) 4598.

7. "Measurement and Interpretation of Counterion Distributions around Cylindrical Micelles,"
S. L. Chang, C. Y. Ku, S. H. Chen, and J. S. Lin,
invited to "Small Angle Scattering," Saclay, France, April 1993. *J. de Physique IV, Col. C8, Suppl. au J. de Phys. I* **3** (1993) 117.

8. "Structure and Dynamics of Dense Water-in-oil Microemulsions below and above the Percolation Threshold,"
S. H. Chen, C. Y. Ku, J. Rouch, P. Tartaglia ,C. Cametti, and J. Samseth,
invited to "Complex Liquid Systems", Calabria, Italy, July 1992. *J. de Physique IV, Col. C1, Suppl. au J. de Phys. II* **3** (1993) 143.

9. "Measurement of Thermal Neutrons Spectrum Using Total Reflection Edge,"
X. L. Zhou and C. Y. Ku,
In preparation. Theoretical assessment and experiments have been done on using the total reflection edge of a super mirror to determine the thermal neutron spectrum of MITR II research reactor. Which is the fundamental research for better utilization of the neutrons produced by a reactor in a spectrometer.

Appendix A

Resolution Function

This is a source code listing of two FORTRAN functions which calculate the resolution function with given parameters.

```
c
      FUNCTION FRES(SIGQ,QCEN,Q)
      X=QCEN*Q/SIGQ
         T=X/3.75
         IF(ABS(X).GT. 3.75) GOTO 100
         BESS=1.+3.5156229*T*T+3.0899424*T**4.+1.2067492*T**6.
     !+0.2659732*T**8.+0.0360768*T**10.+0.0045813*T**12.
      FRESX=Q/SIGQ*EXP(-0.5*(QCEN*QCEN+Q*Q)/SIGQ)*BESS
         GOTO 200
100      BESSX=(0.39894228+0.01328592/T+0.00225319/T**2.                    10
     !-0.00157565/T**3.+0.00916281/T**4.-0.02057706/T**5.
     !+0.02635537/T**6.-0.01647633/T**7.+0.00392377/T**8.)
      FRESX=Q/SIGQ*EXP(-0.5*(QCEN*QCEN+Q*Q)/SIGQ+X)*BESSX/SQRT(X)
200      FRES=FRESX
      RETURN
      END
c
c
     FUNCTION SIGQ(X)
         COMMON/RES_PAR/R1,R2,RL1,RL2,RLAM,DLAM,RDET                        20
     DIMENSION X(1)          ! Array with independent parameters
```

```
      PI=3.1415927

C     R1=0.8       ! RAD IN CM OF SOURCE SLIT
C     R2=0.2       !              SAMPLE
C     RL1=600.     ! DISTANCE IN CM SOURCE-SAMPLE
C     RL2=600.     !                SAMPLE-DETECTOR
C     RLAM=6.00    ! WAVELENGTH IN A
C     DLAM=0.18    ! RELATIVE WAVELENGTH SPREAD OF MONOCHR.                  30
C     RDET=1.0

      WK0=2.*PI/RLAM
      SIGDET=RDET*WK0/RL2*0.4246609

C     SIGRAV=0.0048*0.4246609
      SIGRAV=0.000
C
      THET0=ASIN(X(1)/2./WK0)
      AA1=R1/(RL1+RL2/COS(2.*THET0)**2)                                      40
      A2=R2*COS(2.*THET0)**2./RL2
C
      IF(A2. GT. AA1) GOTO 125
C
      BET1=2.*R1/RL1-0.5*R2*R2*(COS(2*THET0))**4.*(RL1
     !+RL2/COS(2.*THET0)**2)**2./RL1/RL2/RL2/R1
      GOTO 127
C
125   BET1=2.*R2*(1./RL1+COS(2.*THET0)**2/RL2)
     !-0.5*R1*R1*RL2/(R2*COS(2.*THET0)**2.*                                  50
     ! RL1*(RL1+RL2/COS(2.*THET0)**2))
C
C
127   DELQ1=SQRT((WK0*COS(THET0)*BET1)**2.+(X(1)*DLAM)**2.)
CC127 DELQ1=SQRT((WK0*COS(THET0)*BET1)**2.)
C
      SIGQ=(0.4246609*DELQ1)**2.+SIGDET**2.+SIGRAV**2.
```

```
RETURN
END
```

Appendix B

Baxter's Model

This is a source code listing of a FORTRAN program which calculates the structure factor and inter-particle structure of a multi-component droplet system based on Baxter's model.

```
c CORE version
c NEW STICKYNESS PROGRAM LAST UPDATE 17/03/91
c                          12/04/92
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c    This program calculates structure factors or scattering
c    intensities for a system of polydisperse sticky hard
c    spheres. The calculations are based on the theory in the
c    following articles :
c
c    R.J. Baxter              J.Chem.Phys. 49(1968) 2770                  10
c    R.J. Baxter              J.Chem.Phys. 52(1970) 4559
c    J.W.Perram en E.R.Smith  Chem.Phys.Lett. 35(1975) 138
c    A.Vrij                   J.Chem.Phys. 69(1978) 1742
c    P.van Beurten en A.Vrij  J.Chem.Phys. 74(1981) 2744
c
c    The above authors do not agree in their notations.
c    For the particle size the diameter is used but to make
c    life difficult this quantity is called R by Baxter,
c    A by P&S and d by Vrij. In this program it is called r.
```

```
c    q is the scattering vector magnitude (Vrij:K)                    20
c
c    rm  mean diameter
c    tau   the mysterious factor which according to
c          Baxter(1968) is the stickyness / the temperatuur.
c          It is now possible to have polydisperse tau-s!!!
c          This is made possible through the routine set_tau
c          and the parameters atau and btau. (atau=0 gives
c          the monodisperse tau case). In the polydisperse tau
c          case the tau-values are coupled to the particle
c          diameters.                                                  30
c    s     relative measure of size dispersion
c    p     number of classes in size distribution.
c    labda matrix of which the element (i,j) describes the
c         interaction between particles with diam. r(i) and r(j).
c    phi   volumefraction hardspheres. (Baxter 68: eta,
c         Baxter 70, Vrij: ksi3)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c    We are using the following COMMON blocks:
c    [blank] p, atau, btau, tau(200), pi                              40
c    /fu/    k2, k3, h, h2, pi12, pi24
c            constants, belonging to a certain set of
c            phi, rm, tau and sigma
c    /ve/    v(20), la(200), la2(200)
c            index in labda in the 1d-form of labda(i,j)
c    /pa/    x(20), r(20)
c            particle densities and diameters
c    /wl/    i, j
c    /ri/    r1(20,20), r2(20,20), rr(20,20)
c            factors to speed up S(i,j) - calculations                50
c    /be/    b(20)
c            constants in S(i,j)
c    /in/    istat, qa(200), ina(200)
c            a reminder for INTENS
c    /pl/    iplot
```

```
c     /vg/    nvg, vgq(200), vgi(200)
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c ROUTINES CALLED  :
c     calc, show_tau, deal, intens, subint, contact,
c     wijzlab, set_deeltje                                              60
c
c FUNCTIONS CALLED :
c     igetval, agetval
c
      program labda
c
c
      implicit none
      integer    MAXSPEC, MAXPNT
      parameter (MAXSPEC = 2, MAXPNT = 128)                             70
c
c GLOBAL variables :
      integer p, v, li, lj, iplot, istat, nvg, idistr
      real*8  atau, btau, tau, pi, x, x2, vgq, vgi, f_value, dzeep
      integer iform, nrho, nlayer, ina
      real*8  rho(12), dlayer(10), qa
      character*20 lfn
      common          p, idistr, atau, btau, tau(200), pi
      common  /wl/     li, lj
      common  /ve/     v(20), x(200), x2(200)                           80
      common  /pl/     iplot
      common  /in/     istat, qa(200), ina(200)
      common  /nm/      lfn
      common  /vg/     nvg, vgq(200), vgi(200)
      common  /ltrap/  f_value
      common  /deeltje/ iform,rho,nrho,dlayer,nlayer,dzeep
c
c LOCAL variables  :
      integer     i
      real*8      rm, sig, phi, aint0, ai                               90
      character*2  ch
```

```
c
c FUNCTION calls
      real*8  agetval
      integer igetval
c=======================================================================
c     We will first initialize a few things
c
      idistr = 1
      iform=1                                                          100
      rho(1)= 1.0d0
      rho(2)= 0.0d0
      rho(3)= 0.0d0
      dlayer(1) = 0.0d0
      nlayer=0
      nrho=2
c
      pi = 4.0d0 * datan(1.0d0)
c
      rm = 100.0d0                                                     110
      sig = .10d0
      phi = 0.1d0
      p = 9
      atau = 0.0d0
      btau = 99.0d0
      do 5 i=1,100
 5       x(i) = 0.0d0
      li=1
      lj=1
      v(1)=0                                                           120
      do 6 i=1,19
 6       v(i+1) = v(i) + i
c
c     User may mess around now
c
 10   write (*,1000)
      read (*,2000) ch
```

```
      if (ch.eq.'k') goto 99
      if (ch.eq.'h') write (6,1001)
      if (ch.eq.'?t') call show_tau                                    130
      if (ch.eq.'?') then
        write (*,1002) rm/2,sig,phi,
     -       p, idistr,
     -       atau,btau
      endif
      if (ch.eq.'p') then
        p = igetval(1,19)
        if (idistr.eq.3) p = 2
      endif
      if (ch.eq.'id') then                                             140
        idistr = igetval(1,3)
        if (idistr.eq.3) p = 2
      endif
      if (ch.eq.'r') then
c
c     User supplies a radius... Immediately converted to a diameter
c
        rm = 2.0d0*agetval(10.0d0,100.0d0)
c        call calc(rm,sig,phi)
      endif                                                            150
      if (ch.eq.'s') then
        sig = agetval(0.0d0,0.5001d0)
c        call calc(rm,sig,phi)
      endif
      if (ch.eq.'t') then
        write(*,'(A,$)') ' Enter value atau '
        atau = agetval (-1.0d0,1.0d0)
        write(*,'(A,$)') ' Enter value btau '
        btau = agetval (-0.089d0, 99.001d0)
c        call calc(rm,sig,phi)                                          160
      endif
      if (ch.eq.'f') then
        phi = agetval(0.0d0,1.0d0)
```

```
      endif
      if (ch.eq.'l') call wijzlab(p)
      if (ch.eq.'c') then
        call calc(rm,sig,phi)
      endif
      if (ch.eq.'a') call set_deeltje
      if (ch.eq.'i') then                                              170
        call deal(rm,sig,phi,.false.)
        call intens
      endif
      if (ch.eq.'o') then
        call subint(0.0d0,ai,aint0)
        write (*,1007) ai/aint0
      endif
      goto 10
   99 continue
      stop 'Aaaarch!!!'                                                180
 1000 format (' Give instruction :  ',$)
 1001 format (' INSTR name min max        EXPLANATION '/
     -     '================================================='/
     -     '   f   phi_hb 0   1  :  volume fraction inner spheres'/
     -     '   id  idistr 1   3  :  type of distribution'/
     -     '   p   p      3  11  :  number of classes'/
     -     '   r   rm    10 100  :  hard spheres radius'/
     -     '   s   sigma  0  .3  :  relative variance'/
     -     '   t   tau-s  0 100  :  stickiness'/
     -     '================================================='/          190
     -     '   c   calc          :  call calc'/
     -     '   h   help          :  display this menu'/
     -     '   i   intens        :  calculate I(q) or S(q) '/
     -     '   k   kill          :  terminate program'/
     -     '   l   lambda        :  display or change lambda'/
     -     '   o   S(0)          :  calculate S(q=0)'/
     -     '   ?      show          :  display parameters'/
     -     '   ?t  show tau      :  display tau'/
     -     '================================================')
```

```
1002 format (' rm     :',3x,f5.0,'  sigma :',3x,f5.3,'   phi :',f5.3,/,
   -  '  p    :',6x,i2, '  idistr :',6x,i2,/,
   -  ' atau :',f8.3,  '    btau :',f8.3)
1007 format(' S(q=0) = ',e14.5)
2000 format (a)
     end
c
c
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
     subroutine calc(rm,sig,phi)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE: calculate labda(i,j) given:
c              idistr, p, rm, sig, tau(i,j), phi
c
c ROUTINES CALLED:
c    deal, ZSPOW
c
c FUNCTIONS CALLED : none
c
     implicit none
c
c GLOBAL variables :
     integer p, v, idistr
     real*8  atau, btau, tau, pi, f_value
     real*8  x, x2
     common  p, idistr, atau, btau, tau(200), pi
     common /ve/ v(20), x(200), x2(200)
     common /ltrap/ f_value
c
c SUBROUTINE variables :
     real*8  rm, sig, phi
c
c LOCAL variables :
     integer i
c
c ZSPOW variables :
```

```
      integer   ier, n, nsig, itmax
      real      wk(4000), par(1), xla(45)
      real      fnorm
      external  fcn
c=======================================================================
      n = p*(p+1)/2
      call deal( rm,sig,phi,.true.)
      nsig = 5
      itmax = 10
      do 2 i=1,n
        xla(i) = sngl(x(i))
 2    continue
      call ZSPOW (fcn, nsig, n, itmax, par, xla, fnorm, wk, ier)
      f_value = dble(fnorm)
      do 3 i=1,n
 3      x(i) = dble(xla(i))
      if (f_value.gt.1.d-4) write(*,1000) fnorm
 999  return
 1000 format(' No real solution :  ',e10.2)
      end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine deal( rm, sig, phi, labdas )
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE: deals with frequent initialisations.
c
c CALLED BY: calc
c
c ROUTINES CALLED:
c     schultz, setconst
c
      implicit none
c
c GLOBAL variables :
      integer p, v, idistr
      real*8  atau, btau, tau, pi, x, x2
      common  p, idistr, atau, btau, tau(200), pi
```

```
      common  /ve/ v(20), x(200), x2(200)
c
c SUBROUTINE variables :
      real*8  rm, sig, phi
      logical labdas
c
c LOCAL variables :
      real*8  sr, x0
c FUNCTION calls                                                    280
c=====================================================================:
c x0.....
      sr = sig**2
      if (idistr.eq.1) then
         x0 = 6.0d0 * phi / pi / rm**3
     -        / (1.0d0 + 2.0d0 * sr) / (1.0d0 + sr)
      else if (idistr.eq.3) then
         x0 = 6.0d0 * phi / pi / rm**3
     -        / (1.0d0 + 6.0d0 * sr)
      endif                                                         290
c
      call schultz(x0,rm,sig)
      call setconst(labdas)
      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-
      subroutine set_tau
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++-
c PURPOSE: Set values for the stickyness parameter tau. This may be done
c          two ways.                                                300
c       1: through curious function relating the particle diameters to a
c          stickyness
c    or 2: by reading data from a file. (Code is COMMENTED away)
c
c CALLED BY: main, setconst
c
c ROUTINES CALLED: errmsg
```

```
c
      implicit none
c                                                                      310
c GLOBAL variables :
      real*8  x, r, atau, btau, tau, pi
      integer p, idistr
      common  p, idistr, atau, btau, tau(200), pi
      common /pa/ x(20), r(20)
c
c LOCAL variables :
      integer i, j
      real*8  rm, xi
c                                                                      320
cc    character*80  fln, filename
cc    integer jchar, ios, iopt
c
c FUNCTION calls :
      integer inx
cc    integer igets
c=====================================================================
c calculate average diameter
      rm = 0.0d0
      xi = 0.0d0                                                       330
      do 10 i=1,p
        rm = rm + x(i)*r(i)
        xi = xi + x(i)
 10   continue
      rm = rm / xi
c
c
cc100   iopt = 0
cc      write (6,1000) atau,btau
cc      read( 5, fmt='(i1)',end=101,err=100) iopt                      340
cc101   if (iopt.eq.0) iopt=1
cc      if ( iopt .eq. 1 ) then
cc iopt = 1 tau data calculated
```

```
cc
      do 20 i=1,p
20       tau(inx(i,i)) = btau *
     -              ( 1.0d0 + 2.0d0/pi * datan (atau*(r(i)-rm)) )
      do 30 i=1,p
        do 40 j=1,p
          tau(inx(i,j)) = 0.5d0* ( tau(inx(i,i)) + tau(inx(j,j)) )                350
40       continue
30     continue
cc
cc iopt = 2 tau data from file !
cc     else
c%%%%%
cc 50     filename = ' '
cc        write(6,1020)
cc        if (igets(5,filename).eq.0) goto 100
cc        if (filename.eq.'    ') filename='tau.dat'                              360
cc        jchar    = index(filename,' ') - 1
cc        fln(1:jchar) = filename(1:jchar)
cc        open(unit=1,file=fln(1:jchar),blocksize=2048,recl=32766,
cc     -       form='formatted',status='old',iostat=ios)
cc        if (ios.ne.0) then
cc           call errmsg ('Error:  File does not exist')
cc           goto 50
cc        else
cc           write(6,1010) fln(1:jchar)
cc           read(unit=1,fmt=1040) p                                             370
cc           read (unit=1,fmt=*,err=999) ((tau(inx(i,j)),i=1,p),j=1,p)
cc        do 20 i=1,p
cc          do 21 j=1,p
cc            read(unit=1,fmt=*,err=999) tau(inx(i,j))
cc 21        continue
cc 20     continue
cc           close(unit=1)
cc        endif
cc     endif
```

```
      return                                                              380
c999  call errmsg(' Error read tau-data ')
c     close (unit=1)
c     goto 50
1000  format (' Option for tau-data :  ',/,
     -       ' (1) Function with ',/,
     -       '     atau :  ',f5.2,' btau = ',f7.3,/,
     -       ' (2) From File   [1] :  '$)
1010  format(' Reading from datafile :  ',a)
1020  format(' >>> Enter file name or <crtl Z> [tau.dat] :  ',$)
1040  format(i4)                                                          390
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine setconst(labdas)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE: calculate constants belong to a set of x(i) and r(i)
c         results in common blocks  /fu/, /ri/, /ve/, /be/
c
c CALLED BY: deal
c
c ROUTINES CALLED  : set_tau                                              400
c
c FUNCTIONS CALLED : fksi, inx
c
      implicit none
c
c GLOBAL variables :
      real*8  atau, btau, tau, pi, k2, k3, h, h2, pi12, pi6
      real*8  x, r, r1, r2, rr, la, l2, b
      integer p, v, idistr
      common /fu/ k2, k3, h, h2, pi12, pi6                                410
      common p, idistr, atau, btau, tau(200), pi  /pa/ x(20), r(20)
      common /ri/ r1(20,20), r2(20,20), rr(20,20)
      common /ve/ v(20),la(200),l2(200) /be/ b(20)
c
      logical labdas
```

```
c
c LOCAL variables :
      real*8  bla, dla, b0, alab, som
      integer i, j
c FUNCTION calls                                                    420
      real*8  fksi
      integer inx
c=====================================================================:
      k2 = fksi(2)
      k3 = fksi(3)
      h = 1.0d0 - k3
      h2 = h**2
      pi12 = pi/12.0d0
      pi6 = pi/6.0d0
      do 200 i=1,p                                                  430
        do 100 j=1,p
          r1(i,j) = 0.5d0*(r(i)+r(j))
          r2(i,j) = r1(i,j)**2
          rr(i,j) = r(i)*r(j)
 100    continue
 200  continue
c
c r(i)'s are calcultated now calculate tau-array
c
      call set_tau                                                  440
      if (labdas) then
          bla = k3 / h + tau(inx(1,1))
          dla = bla**2 - k3 / 3.0d0 * (1.0d0 + 0.5d0*k3) / h2
          if (dla.ge.0.0d0) then
             alab = 6.0d0 / k3 * ( bla -dsqrt(dla) )
          else
             alab = 6.0d0 / k3 * bla
          endif
          do 300 i=1,200
             la(i) = alab                                           450
 300      continue
```

```
      endif
c
      do 400 i=1,p
         som = 0.0d0
         do 350 j=1,p
              som = som +
     -              x(j) * la(inx(i,j)) * r2(i,j) * r(j)
 350     continue
         b0 = 3.0d0*r(i)*k2/h2 - pi6*som/h                                    460
         b(i) = -0.5d0*b0*r(i)
 400  continue
      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine schultz(x0,rmean,sig)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE :
c     The number densities of particles with diameter r(i) are given
c     by the size distribution (determined by 'idistr')                    470
c -->    result calculation passed to x(i) and r(i) in /pa/
c
c !!!  Normal distribution not yet implemented !!!
c idistr :
c  1  Schulz
c  2  Normal
c              The diameter run between plus
c               and minus three times sigma of the mean.
c  3  Bimodal
c                                                                          480
c CALLED BY : deal
c
c ROUTINES CALLED : none
c
c INPUT
c -------
c x0    :  normalisation ie  phi = ksi(3).
```

```
c rmean :  mean of the distribution.
c sig   :  relative variance of the distribution.
c                                                                      490
      implicit none
      real*8     del
      parameter (del=3.0d0)
c
c GLOBAL variables :
      integer p, idistr
      real*8  atau, btau, tau, pi
      real*8  x, r1
      common p, idistr, atau, btau, tau(200), pi
      common /pa/ x(20), r1(20)                                        500
c        r1(i) is everywhere else r(i)
c
c SUBROUTINE variables :
      real*8  x0, rmean, sig
c
c LOCAL variables :
      integer i
      real*8  zp, dr, r, fac, rm, z
c
c FUNCTION calls :                                                     510
      real*8  sch, dgamma
c dgamma :  library gamma-function
c sch    :  schulz distribution function
c=======================================================================
      rm=rmean
c
c pathological cases p=1 or sig = 0
      if (p.eq.1 .or.  sig.eq.0.0) then
         do 11 i=1,p
           r1(i)=rm                                                    520
           x(i)=x0/dfloat(p)
 11      continue
         return
```

```
      endif
c
c more interesting cases p > 1 , sig <> 0
c
      if (idistr.eq.1) then
c SCHULZ option
c normalisation of Schulz distributionfunction                          530
c
c                        (Z+1)
c                    (Z+1)
c     fac =      --------------------
c              Gamma(Z+1) * rm * exp(Z+1)
c
      z = (1.0d0/sig)**2 - 1.0d0
      zp = z + 1.0d0
      if (sig.lt.0.25d0) then
         fac = dsqrt (zp/2.0d0/pi) / rm /(1.0d0 + 1.0d0/12.d0/zp          540
     -             + 1.0d0/2.88d2/zp**2 - 1.39d2/5.1840d4/zp**3
     -            - 571.0d0/2.488320d6/zp**4)
c approximation of Gamma function for large arguments
c        Zie Handbook of Chemistry and Physics, Gamma function
      else
c
c exact expression for Gamma function for small arguments
         fac = dexp( zp*dlog(zp) - zp )/rm/dgamma(zp)
      endif
c                                                                       550
c !!!  for large sigma distribution is skewed --->
c
      dr = 2.0d0 * del * sig * rm/ dfloat(p-1)
      r = rm - del * sig * rm
      if (r.lt.0.0d0) r = dr/2.0d0
         do 200 i=1,p
            r1(i) = r
            x(i) = x0 * sch(r, fac, rm, z) * dr
            r = r + dr
```

```
 200     continue                                                        560
      return
c end of SCHULZ option
      endif
c
c      if (idistr.eq.2) then
c NORMAL distribution
c end of NORMAL distribution
c      return
c      endif
c                                                                       570
      if (idistr.eq.3) then
c BIMODAL option
         x(1) = 0.5d0 * x0
         x(2) = x(1)
         r1(1) = rm - rm * sig
         r1(2) = rm + rm * sig
c end of BIMODAL option
         return
      endif
      end                                                               580
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      real*8  function sch(r, fac, rm, z)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE :
c     Calculate given fac, rm & z the
c     value of the schultz distribution function f(x) at x = r
c
c CALLED BY :  schultz
c
      implicit none                                                     590
c
c SUBROUTINE variables :
      real*8  r, fac, rm, z
c
c LOCAL variables  :
```

```
      real*8  rf
c=======================================================================
      rf = r / rm
      sch = fac * dexp (z * dlog(rf) + (z+1.0d0) * (1.0d0-rf))
      return                                                          600
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      real*8 function fksi(n)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE: Calculate worldfamous ksi2 and ksi3 from x(i) and r(i), i=1..p
c     (See:  Vrij 1979 (17) )
c
c CALLED BY: setconst
c
c ROUTINES CALLED: none                                               610
c
      implicit none
c
c GLOBAL variables :
      integer   p, idistr
      real*8    atau, btau, tau, pi, x, r
      common p, idistr, atau, btau, tau(200), pi /pa/ x(20),r(20)
c
c SUBROUTINE variables :
      integer n                                                       620
c
c LOCAL variables :
      integer i
c=======================================================================
      fksi=0.0d0
      do 100 i=1,p
 100    fksi=fksi+x(i)*r(i)**n
      fksi=fksi*pi/6.0d0
      return
      end                                                             630
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
      subroutine fcn(xla, f, n, par)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE :
c    This function is the obedient victim of the IMSL routine ZSPOW
c    It is a sad compromise between readability and speed.
c    In f(i) the Perram&Smith 1975 (14) equations are held.
c
c CALLED BY :
c    ZSPOW ( in calc )                                                    640
c
c ROUTINES CALLED :  none
c
c FUNCTIONS CALLED : inx
c
c GLOBAL variables :
      integer p, idistr
      real*8  x, r, r1, r2, rr, b
      real*8   atau, btau, tau, pi, k2, k3, h, h2, pi12, pi6
      common p, idistr,  atau, btau, tau(200), pi /pa/ x(20), r(20)        650
      common /ri/ r1(20,20), r2(20,20), rr(20,20)
      common /fu/ k2, k3, h, h2, pi12, pi6
      common /be/ b(20)
c
c SUBROUTINE variables :
      real    xla(45), f(45), par(1)
c      real    aux(15)
      integer n
c
c LOCAL variables  :                                                       660
      real*8  xc(20), a(20)
      real*8  som, b0
      integer iq
c FUNCTION calls
      integer inx
c=============================================================================
c     In the following loop the coefficients as used in Perram&Smith
```

```
c      1975 (9) are calculated.  The b(i)'s are put in a common block
c     as these are used later for calculation of the structure factor.
c                                                                       670
      do 200 i=1,p
        som = 0.0d0
        do 100 j=1,p
           som = som +
     -          x(j) * dble(xla(inx(i,j))) * r2(i,j) * r(j)
  100    continue
        xc(i) = som*pi6
        b0 = 3.0d0*r(i)*k2/h2 - xc(i)/h
        a(i) = 1.0d0/h + b0
        b(i) = -0.5d0*b0*r(i)                                           680
  200 continue
c------
c
c     write(*,1015)
c 1015 format('$Concentrations :  ')
c     write(*,1016) (x(i),i=1,5)
c 1016 format(' ',5e14.7)
c     write(*,1018)
c 1018 format('$Radii          :  ')
c     write(*,1019) (r(i),i=1,5)                                        690
c 1019 format(' ',5e14.7)
      iq = 0
      do 500 i=1,p
        do 400 j=1,i
           som = 0.0d0
           do 300 k=1,p
              som = som + x(k) * dble(xla(inx(j,k)))
     -             * r2(j,k) / r1(i,j) *
     -             (
     -               -0.5d0*a(i)*rr(i,k) - b(i)*r(k)                  700
     -               + dble(xla(inx(i,k)))*r2(i,k)/12.0d0
     -             )
  300        continue
```

```
            iq = iq + 1
            f(iq) = sngl( som*pi6 + b(i)/r1(i,j) + a(i)
     -                  - tau(inx(i,j)) * dble(xla(inx(i,j))) )
 400     continue
 500  continue
c      write(*,1004) n
c     do 15 ih=1,5                                                    710
c       do 14 jh=1,5
c 14          aux(jh)=xla(inx(ih,jh))
c             write (*,1013) (aux(jh),jh=1,5)
c 1013              format (' ',5e14.6)
c 15  continue
c      write (*,1003)
c      write (*,1002) (f(i),i=1,15)
c 1002 format(' ',5e14.7,/,' ',5e14.7,/,' ',5e14.7)
c 1003 format('$Equations :  ')
c 1004 format('$labdas :  ',i4)                                        720
      return
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      integer function inx(i,j)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c     Omdat E04CGF insisteert op een array van variabelen, terwijl
c     labda toch een heuse matrix is, simuleert inx een pxp-matrix.
c     Het array v bevat offsets zodat de symmetrie behouden blijft.
c     B.v. v1=0, v2=1, v3=3 etc. (Zie SUBROUTINE setconst)
c                                                                      730
      implicit none
c
c GLOBAL variables :
      integer v
      real*8  la, l2
      common /ve/ v(20), la(200), l2(200)
c
c SUBROUTINE variables :
      integer i, j
```

```
c=============================================================740==========
      if (i.lt.j) goto 100
      inx = v(i)+j
      return
 100  inx = v(j)+i
      return
      end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine wijzlab(p)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE : mess-around and display current values of labda(i,j)          750
c
c CALLED BY: main
c
c ROUTINES CALLED : none
c
      implicit none
c
c GLOBAL variables :
      integer v, i, j
      real*8  la, l2                                                      760
      common /wl/ i, j /ve/ v(20), la(200), l2(200)
c
c SUBROUTINE variables :
      integer p
c
c LOCAL variables :
      character*1 ch
      real*8  aux(9)
      integer ih, jh
c                                                                         770
c FUNCTION calls
      integer inx, igetval
      real*8  agetval
c=============================================================
 10   write (*,1000) i,j,la(inx(i,j))
```

```
      read(*,1001) ch
      if (ch.eq.'i') i = igetval(1,p)
      if (ch.eq.'j') j = igetval(1,p)
      if (ch.eq.'l') la(inx(i,j))=agetval(0.0d0,30.0d0)
      if (ch.eq.'q') return                                              780
      if (ch.ne.'?') goto 20
      do 15 ih=1,p
        do 14 jh=1,p
 14          aux(jh)=la(inx(ih,jh))
        write (*,1003) (aux(jh),jh=1,p)
 15   continue
 20   continue
      goto 10
 1000 format (' i =',i2,', j =',i2,', l(i,j) =',f5.2,' ?  '$)
 1001 format (a)                                                         790
 1003 format (' ',9f6.2)
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine show_tau
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE : display current values of tau(i,j)
c
c CALLED BY: main
c
c ROUTINES CALLED : none                                                 800
c
      implicit none
c
c GLOBAL variables :
      real*8  atau, btau, tau, pi
      integer  p, idistr
      integer v, i, j
      real*8  la, l2
      common  p, idistr, atau, btau, tau(200), pi
      common /wl/ i, j /ve/ v(20), la(200), l2(200)                      810
c
```

```
c LOCAL variables :
      real*8  aux(9)
      integer ih, jh
c FUNCTION calls
      integer inx
c=======================================================================
        do 15 ih=1,p
          do 14 jh=1,p
 14           aux(jh)=tau(inx(ih,jh))                                    820
        write (*,1003) (aux(jh),jh=1,p)
 15     continue
 1003 format (9(1x,f6.2))
      end
c
c
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine struct(q,s,ifail)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE:                                                               830
c   S is the inverse of E-C~(q) [Vrij 1979 (8)] and can be determined
c   from Q~. As Q~(-q)=Q~*(q) (*=complex conjugated) we find
c   E-C~(q) = Q~+(q)Q~(q) (Q~ times its hermitian conjugated). The
c   inverse is determined by the modified to double precision IMSL
c   routine D_LEQT1D.
c
c CALLED BY :
c     subint
c
c ROUTINES CALLED  :                                                     840
c     D_LEQT1D, qik
c
      implicit none
c
c GLOBAL variables :
      integer p, idistr
      real*8  atau, btau, tau, pi
```

```
      common  p, idistr, atau, btau, tau(200), pi
c
c SUBROUTINE variables :                                              850
      real*8  q
      integer ifail
      complex*16   s(20,20)
c
c LOCAL variables  :
      complex*16   qq(20,20)
      complex*16   A(20,20), B(20,20), som
      real*8       WA(20)
      integer      ier, N, M, IA, IB, IJOB
      integer      i, j, k                                            860
c=======================================================================
      ifail = 0
      call qik(q,qq)
      do 300 i=1,p
         do 200 j=1,p
           A(i,j) = qq(i,j)
           B(i,j) = dcmplx ( 0.0d0, 0.0d0 )
           if (i.eq.j) B(i,j) = dcmplx ( 1.0d0 , 0.0d0 )
 200     continue
 300  continue                                                         870
c
      N = p
      IA = 20
      IB = 20
      M = p
      IJOB = 0
      CALL D_LEQT1C (A,N,IA,B,M,IB,IJOB,WA,IER)
      if (ier.ne.0) ifail = 1
c Q
      do 400 i=1,p                                                     880
        do 500 j=1,p
          som = (0.0d0,0.0d0)
          do 600 k =1,p
```

```
            som = som + B(i,k) * dconjg( B(j,k) )
 600      continue
c           write (6,*) i,j,som
          s(i,j) = som
 500    continue
 400  continue
      return                                                          890
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine subint(q,ai,aint0)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE :
c     Calculate scattering value including interactions (ai) for the
c     system at scat.vector=q, and excluding interactions (aint0)
c
c !! NOTE !! in principle different particle electron density profiles
c can be introduced through the function 'form_factor'. Caution should          900
c be taken here as strange results may be introduced. (negative overall
c particle radii)
c For homogenous particles IFORM=1 (and dlayer(i) = 0 and dzeep=0)
c no problems arise.
c
c ROUTINES CALLED :
c     struct
c
c FUNCTIONS CALLED :
c     form_factor                                                       910
c
      implicit none
c
c GLOBAL variables :
      integer p, idistr, istat, nrho, nlayer, iform
      real*8  atau, btau, tau, pi, dzeep
      real*8  x, r, rho(12), dlayer(10), qa, ina
      common  p, idistr, atau, btau, tau(200), pi
      common /pa/ x(20), r(20) /in/ istat, qa(200), ina(200)
```

```
      common / deeltje / iform,rho,nrho,dlayer,nlayer,dzeep                920
c
c SUBROUTINE variables :
      real*8  q, ai, aint0
c
c LOCAL variables :
      real*8       a(20), p43
      complex*16   s(20,20)
      integer      i, j, ifail
c FUNCTION calls
      real*8  form_factor                                                  930
c=====================================================================
      aint0 = 0.0
      if (q.lt.0.0001) then
        p43 = 4.0d0 * pi / 3.0d0
        do 5 i=1,p
           a(i) = p43 * r(i)**3
           aint0 = aint0 + a(i)**2 * x(i)
5       continue
      else
        do 10 i=1,p                                                        940
           a(i) = form_factor(iform, q,
     -             r(i)*0.5d0 - dzeep - dlayer(1),
     -             nrho, rho, nlayer, dlayer)
           aint0 = aint0 + a(i)**2 * x(i)
10      continue
      endif
      call struct(q,s,ifail)
      if (ifail.ne.0) then
        if (ifail.eq.1) write (*,'(a)') '$SNAG: Complex diagonal.'
        if (ifail.eq.2) write (*,'(a)') '$SNAG: QQ+ is not pos.  def.'     950
        write (*,'(a)') '$Intens Abort - You lose.'
        return
      endif
110   continue
      ai = 0.0d0
```

```
c     --- intens ---
      do 300 i=1,p
        do 200 j=i+1,p
            ai = ai + 2. * dreal(s(i,j))
     -              * a(i) * a(j) * dsqrt(x(i)*x(j))
c               As S is hermitian, you can write
c               the sum this way.
 200    continue
        ai = ai + a(i)**2 * dreal(s(i,i)) * x(i)
c               s(i,i) is real, but the machine
c               does not know.
 300  continue
      return
      end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine qik(q,qq)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE:
c     This routine determines Q~(i,k) as defined in Vrij 1979 (23) but now
c     with stickyness included. For insiders we note that the variables
c     start with vr.
c
c CALLED BY: struct
c
c ROUTINES CALLED: none
c
      implicit none
c
c GLOBAL variables :
      integer p, v, idistr
      real*8  atau, btau, tau, pi, k2, k3, h, h2, pi12, pi6
      real*8  la, l2, x, r, b
      common p, idistr, atau, btau, tau(200), pi
      common /ve/ v(20), la(200), l2(200)
      common /pa/ x(20), r(20) /fu/ k2, k3, h, h2, pi12, pi6
      common /be/ b(20)
```

```
c
c SUBROUTINE variables :
      real*8  q
      complex*16   qq(20,20)
c
c LOCAL variables  :
      integer  i, k
      real*8   sq(20), cq(20), vrx(20), vrphi(20), vrpsi(20)
c FUNCTION calls                                                    1000
      integer inx
c=====================================================================:
      if (q.lt.0.0001d0) then
        do 30 i=1,p
          vrx(i)=0.0d0
          sq(i)=0.0d0
          cq(i)=1.0d0
          vrphi(i)=1.0d0
          vrpsi(i)=1.0d0
 30     continue                                                    1010
      else
        do 10 i=1,p
          vrx(i) = r(i) * 0.5d0 * q
          sq(i) = dsin(vrx(i))
          cq(i) = dcos(vrx(i))
          vrphi(i) = 3.0d0 / vrx(i)**3
     -          * ( sq(i) - vrx(i) * cq(i) )
          vrpsi(i) = sq(i) / vrx(i)
 10     continue
      endif                                                         1020
c
      do 200 i=1,p
        do 100 k=1,p
          qq(i,k) = dcmplx ( pi6 * dsqrt(x(i)*x(k)) , 0.0d0 )
     -          * dcmplx
     -          (
     -            - 0.25d0 * la( inx(i,k) ) * r(k)
```

```
     -             * ( r(i) + r(k) )**2 * vrpsi(k)
c
c                  This term is obtained by treating Perram&Smith
c                  1975 (9) the way Vrij 1979 (14) does
c
     -            + r(k)**3 / h * vrphi(k)
     -            + r(k)**2 * r(i) / h * 3.0d0 * vrpsi(k)
     -            - r(k)**3 * b(i) / r(i)
     -              * 2.0d0 * vrphi(k)
     -           ,
     -            - r(i) * r(k)**2 / h * vrx(k) * vrphi(k)
     -           )
     -           * dcmplx( cq(i) , sq(i) )
c                 = cdexp(vrx(i))
 100    continue
c                 Now we've got Z(i,k)....
          qq(i,i) = qq(i,i) + dcmplx ( 1.0d0, 0.0d0 )
c                                                   ....  and Q~(i,k)!
 200  continue
      return
      end
c
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine intens
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE :
c     Calcute scattering intensity or structure factor.
c     Write to file.
c
c ROUTINES CALLED :
c     subint, schrijf
c
c
c GLOBAL variables :
      integer p, idistr, istat, iplot
      real*8   atau, btau, tau, pi
```

```
      real*8  x, r, ina, qa
      common p, idistr, atau, btau, tau(200),pi /pa/ x(20),r(20)
      common /pl/ iplot /in/ istat, qa(200), ina(200)
c
c LOCAL variables  :
      real*8       a(20)
      real*8       q, dq, smint, ai, aint0                          1070
      real         xxx(200),yyy(200)
      character*1  ch
      complex*16   s(20,20)
c
c FUNCTION calls :
      integer igets
c=======================================================================
 1    write (*,1000)
      read (*,*) q
      if (q.ne.0.0) goto 10                                         1080
      write (*,1030)
      read (*,1031) ch
      if (ch.eq.'s') istat=1
      if (ch.eq.'i') istat=0
      if (ch.eq.'o') goto 950
      if(.true.)  goto 1
 10   continue
      write (*,1002)
      read (*,*) dq
      write (*,1004)                                                1090
      read (*,*) nq
      smint = 0.0d0
      do 900 n = 1,nq
         call subint (q,ai,aint0)
         qa(n)=q
         if (istat.eq.1) goto 500
         ina(n) = ai
         if (ina(n).lt.smint) smint = ina(n)
         goto 800
```

```
 500     continue                                                        1100
c         --- struct ---
         ina(n) = ai / aint0
         if (ina(n).lt.smint) smint = ina(n)
 800     continue
         q = q + dq
 900  continue
c     do 910 n=1,nq
c        ina(n)=ina(n)-smint
c 910 continue
 950  continue                                                          1110
 960  continue
c      write (6,1017)
c      read (5,1031) ch
c      if (ch.eq.'j') then
c         do 8888 i=1,nq
c            xxx(i) = sngl(qa(i))
c            yyy(i) = sngl(ina(i))
c8888    continue
c         iplot = 0
c         call ctermplot(xxx,yyy,nq,iplot,' ',' ')                      1120
c      else if (ch.ne.'n')then
c         goto 960
c      endif
      call schrijf(nq, qa, ina)
      return
 1000 format (' Give first q: '$)
 1002 format (' Give stepsize: '$)
 1004 format (' Give number I(q)-s: '$)
 1017 format (' Plot? [j/n] '$)
 1030 format (' s(truct)/i(ntens)/o(again): '$)                          1130
 1031 format (1a)
      end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      integer function igetval(il,ih)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

```
c       implicit none
        integer il, ih
c=======================================================================
 11     write (*,20)
 10     read (*,fmt=*,err=11) igetval                                    1140
        if((igetval.ge.il).and.(igetval.le.ih)) return
        write (*,30) il,ih
        if (.true.)  goto 10
 20     format (' Value : '$)
 30     format (' ',i3,' < value < ',i4,'. '/' Again : '$)
        end
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        real*8 function agetval(bl,bh)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c                                                                        1150
        real*8  bl, bh
c=======================================================================
 11     write (*,20)
 10     read (*,fmt=*,err=11) agetval
        if((agetval.ge.bl).and.(agetval.le.bh)) return
        write (*,30) bl,bh
        if (.true.)  goto 10
 20     format (' Value: '$)
 30     format (' ',f6.2,' <= value <= ',f6.2,' . '/' Again : '$)
        end                                                              1160
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
         REAL*8 FUNCTION FSPHERE(DELRHO,R,Q)
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C FUNCTION CALCULATES THE SCATTERED AMPLITUDE FROM A HOMOGENEOUS
C SPHERE OF RADIUS R AT SCATTERING VECTOR Q
C
C GLOBAL :
        REAL*8  DELRHO, R, Q
C LOCAL  :
        REAL*8  PI, RQ                                                   1170
C
```

```
      PI = 4.0D0 * DATAN (1.0D0)
      RQ = R * Q
      IF (Q.LT.1.0D-7) THEN
         FSPHERE=4.0D0*PI*DELRHO*R*R*R/3.0D0
      ELSE
         FSPHERE=4.0D0*PI*DELRHO*(DSIN(RQ)-RQ*DCOS(RQ))/(Q*Q*Q)
      ENDIF
      RETURN
      END                                                                    1180
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      REAL*8  FUNCTION  FORM_FACTOR(IFORM,Q_VECTOR,RADIUS,NRHO,RHO,
     >                          NLAYER,DLAYER)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C       IMPLICIT NONE
C NRHO,NLAYER: Number of RHO(),DLAYER()
C RHO(I)      : Electron densities (MAX = 12)
C DLAYER(I)   : Transition layer distances (MAX = 10)
C
C    Q_VECTOR     : SCATTERING ANGLE                                        1190
C    RADIUS       : MEAN OF THE GAUS DISTRIBUTION
C
C  ROUTINES CALLED :
C ==================
C                 SCATTERED AMPLITUDE (NOT INTENSITY) OF
C   FSPHERE    :  HOMOGENOUS SPHERE
C   FLINSPHERE :  DIFFUSE SPHERE (LINEAR DECREASING DENSITY)
C   FSHELL     :  SHELL
C
      INTEGER IFORM, NRHO, NLAYER                                            1200
      REAL*8  Q_VECTOR, RADIUS
      REAL*8  DLAYER(10), RHO(12)

      REAL*8  FSPHERE,FLINSPHERE,FSHELL
C LOCAL  :
C
      REAL*8    PI, PPI, Y, YZERO
```

```
C=======================================================================
      PI=4.0D0*DATAN(1.0D0)
      PPI=4.0D0*PI/3.0D0                                                1210
      IF (IFORM.EQ.1) THEN
      Y=FSPHERE(RHO(1)-RHO(2),RADIUS,Q_VECTOR)
      YZERO = (RHO(1)-RHO(2))*(RADIUS**3)
      ELSE IF (IFORM.EQ.2) THEN
      Y=FLINSPHERE(RHO(1)-RHO(2),RADIUS,DLAYER(1),Q_VECTOR)
      YZERO = (RHO(1)-RHO(2))*RADIUS**3 +
     >    0.75E0*(RHO(1)-RHO(2))/DLAYER(1) * ((RADIUS+DLAYER(1))**4
     >    -RADIUS**4) + (RHO(1)-RADIUS*(RHO(1)-RHO(2))/DLAYER(1))*
     >    ((RADIUS+DLAYER(1))**3-RADIUS**3)
      ELSE IF (IFORM.EQ.3) THEN                                         1220
      Y=FSHELL(RHO(1)-RHO(2),RHO(2)-RHO(3),RADIUS,
     >          DLAYER(1),Q_VECTOR)
      YZERO = RADIUS**3*(RHO(1)-RHO(2))+
     >                (RADIUS+DLAYER(1))**3*(RHO(2)-RHO(3))
      ELSE IF (IFORM.EQ.4) THEN
      Y=FLINSPHERE(RHO(2)-RHO(3),RADIUS,DLAYER(1),Q_VECTOR)
     >    -FSPHERE(RHO(2)-RHO(1),RADIUS,Q_VECTOR)
      YZERO = (RHO(1)-RHO(2))*RADIUS**3 + (RHO(2)-RHO(3))*RADIUS**3
     >    0.75E0*(RHO(2)-RHO(3))/DLAYER(1) * ((RADIUS+DLAYER(1))**4
     >    -RADIUS**4) + (RHO(2)-RADIUS*(RHO(2)-RHO(3))/DLAYER(1))*    1230
     >    ((RADIUS+DLAYER(1))**3-RADIUS**3)
      ELSE IF (IFORM.EQ.5) THEN
      Y=FLINSPHERE(RHO(1)-RHO(2),RADIUS,DLAYER(1),Q_VECTOR)
     >    +FSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1),Q_VECTOR)
      YZERO = (RHO(1)-RHO(2))*RADIUS**3 +
     >    0.75E0*(RHO(1)-RHO(2))/DLAYER(1) * ((RADIUS+DLAYER(1))**4
     >    -RADIUS**4) + (RHO(1)-RADIUS*(RHO(1)-RHO(2))/DLAYER(1))*
     >    ((RADIUS+DLAYER(1))**3-RADIUS**3)+
     >    (RHO(2)-RHO(3))*(RADIUS+DLAYER(1))**3
      ELSE IF(IFORM.EQ.6) THEN                                          1240
      Y=FSPHERE(RHO(1)-RHO(2),RADIUS,Q_VECTOR)+
     >FLINSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1),DLAYER(2),Q_VECTOR)
      YZERO=1.0D0
```

```
      ELSE IF(IFORM.EQ.7) THEN
      Y=FLINSPHERE(RHO(1)-RHO(2),RADIUS,DLAYER(1),Q_VECTOR)+
     >FLINSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1),DLAYER(2),Q_VECTOR)
      YZERO=1.0D0
      ELSE IF(IFORM.EQ.8) THEN
      Y=FLINSPHERE(RHO(1)-RHO(2),RADIUS,DLAYER(1),Q_VECTOR)+
     >FSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1)+DLAYER(2),Q_VECTOR)                1250
      YZERO=1.0D0
      ELSE IF(IFORM.EQ.9) THEN
      Y=FLINSPHERE(RHO(1)-RHO(2),RADIUS,DLAYER(1),Q_VECTOR)+
     >FLINSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1)+DLAYER(2),
     >           DLAYER(3),Q_VECTOR)
      YZERO=1.0D0
      ELSE IF (IFORM.EQ.10) THEN
      Y=FSPHERE(RHO(1)-RHO(2),RADIUS,Q_VECTOR)+
     >     FSPHERE(RHO(2)-RHO(3),RADIUS+DLAYER(1),Q_VECTOR)+
     >     FSPHERE(RHO(3)-RHO(4),RADIUS+DLAYER(1)+DLAYER(2),Q_VECTOR)           1260
      YZERO = (RHO(1)-RHO(2))*(RADIUS**3)+
     >           (RHO(2)-RHO(3))*((RADIUS+DLAYER(1))**3)+
     >           (RHO(3)-RHO(4))*((RADIUS+DLAYER(1)+DLAYER(2))**3)
      ENDIF
      FORM_FACTOR = Y
C      FORM_FACTOR = Y / YZERO / PPI
      RETURN
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
       REAL*8 FUNCTION FLINSPHERE(DELRHO,R,D,Q)                                1270
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C        IMPLICIT NONE
C FUNCTION CALCULATES THE SCATTERED AMPLITUDE FROM A HOMOGENEOUS
C SPHERE OF RADIUS R , WITH A LINEAR DECREASING ELECTRON DENSITY
C OVER A DISTANCE D , AT SCATTERING VECTOR Q
C
      REAL*8  DELRHO, R, D, Q
      REAL*8  X, Y, PI
        PI=4.0D0*DATAN(1.0D0)
```

```
        X=R*Q                                                                  1280
        Y=(R+D)*Q
      FLINSPHERE= 4.0D0 * PI * DELRHO / (Q*Q*Q) *
     >        ((2.0D0*(DCOS(Y)-DCOS(X))-X*DSIN(X)+Y*DSIN(Y))/(X-Y))
      RETURN
      END
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
        REAL*8 FUNCTION FSHELL(DELRH1,DELRH2,R,D,Q)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C        IMPLICIT NONE
C FUNCTION CALCULATES THE SCATTERED AMPLITUDE FROM A SHELL PARTICLE            1290
C WITH INNER RADIUS R, OUTER RADIUS R+D AT SCATTERING VECTOR Q.
C
      REAL*8  DELRH1,DELRH2,R,D,Q
C
      REAL*8  FSPHERE
C
      FSHELL= FSPHERE(DELRH1,R,Q) + FSPHERE(DELRH2,R+D,Q)
      RETURN
      END
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 1300
       subroutine print_profile(iform)
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c       implicit none
       integer iform
c
       write(*,99) iform
       goto (1,2,3,4,5,6,7,8,9,10) iform
 1     write(*,101)
       return
 2     write(*,102)                                                            1310
       return
 3     write(*,103)
       return
 4     write(*,104)
       return
```

```
 5      write(*,105)
        return
 6      write(*,106)
        return
 7      write(*,107)                                                    1320
        return
 8      write(*,108)
        return
 9      write(*,109)
        return
 10     write(*,110)
        return
  99    format(1h ,/,'$Density profile : iform = ',i4)
 101    format(1h ,/,"----",/,"    |",/,"    |----"$)
 102    format(1h ,/,'   ',/,"---- /",/,"      /",/,"       /----"$)     1330
 103    format(1h ,/,"     ---",/,"    |   |",/,"----|   |",/,
     > "        |----"$)
 104    format(1h ,/,"----",/,"    /",/,"     /----"$)
 105    format(1h ,/,'Not yet drawn?'$)
 106    format(1h ,/,"     ----",/,"    |    /",/,"----|     /",/,
     > "           /-----"$)
 107    format(1h ,/,"      ---",/,"     /   |",/,"----/    |",/,
     > "         |-----"$)
 108    format(1h ,/,"     //",/,"----/  /",/,"        /-----"$)
 109    format(1h ,/,"      ___",/,"     /   /",/,"----/     /",/,       1340
     > "           /----"$)
 110    format(1h ,/,'Not yet drawn?'$)
        end
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine set_deeltje
c++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c      implicit none
c
c GLOBAL :
      integer nrho, nlayer, iform                                       1350
      real*8  rho(12), dlayer(10)
```

```
      real*8  dzeep
      common / deeltje / iform, rho, nrho, dlayer, nlayer, dzeep
c
c LOCAL  :
      integer i
c
c FUNCTION calls
      real*8  agetval
      integer igetval                                                    1360
c=======================================================================
  99  WRITE (*,102)
 102  FORMAT('$Give IFORM : '$)
      IFORM = IGETVAL(1,11)
      IF (IFORM.EQ.1) THEN
         RHO(1) = 1.0D0
         RHO(2) = 0.0D0
         DO 1002 I=1,10
1002        DLAYER(I) = 0.0D0
         NRHO = 2                                                        1370
         NLAYER = 0
      ELSE IF (IFORM.EQ.2) THEN
         NRHO = 2
         NLAYER = 1
      ELSE IF ((IFORM.GE.3).AND.(IFORM.LE.5)) THEN
         NRHO = 3
         NLAYER = 1
      ELSE IF ((IFORM.GE.6).AND.(IFORM.LE.8)) THEN
         NRHO = 3
         NLAYER = 2                                                      1380
      ELSE IF (IFORM.EQ.9) THEN
         NRHO = 3
         NLAYER = 3
      ELSE IF (IFORM.EQ.10) THEN
         NRHO = 4
         NLAYER = 2
      ENDIF
```

```
      WRITE (*,103)
  103 FORMAT('$Give inner density : '$)
      RHO(1) = AGETVAL(0.d0,1000.d0)                                    1390
  104 FORMAT('$Give outer density : '$)
      WRITE (*,104)
      RHO(NRHO) = AGETVAL(-1000.d0,1000.d0)
      DO 98 I=2,NRHO-1
       WRITE(*,100) I-1
  100  FORMAT(1H ,'$ Give electron density in transition layer no.',I4)
       RHO(I) = AGETVAL(-1000.d0,1000.d0)
   98 CONTINUE
      DO 97 I=1,NLAYER
       WRITE(*,101) I                                                   1400
  101  FORMAT(1H ,'$ Give transition layer thickness no.',I4)
       DLAYER(I) = AGETVAL(0.d0,1000.d0)
   97 CONTINUE
      IF(.TRUE.) GOTO 999
      DO 95 I=1,10
   95   CALL PRINT_PROFILE(I)
      IF (.TRUE.)  GOTO 99
  999 RETURN
      END
c                                                                       1410
c
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      subroutine schrijf(nnn, xxx, yyy)
c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
c PURPOSE : save data xxx and yyy in a file
c
c ROUTINES CALLED  :
c     errmsg
c
c FUNCTIONS CALLED :                                                    1420
c     askno, igets
c
c      implicit none
```

```
      real*8        xxx(1), yyy(1)
      integer       nnn
c LOCAL:
c
      character*80  dumfln, infln
      integer       jchar, ios, irc
      logical       answer                                          1430
c
c FUNCTION calls:
      logical       askno
      integer       igets
c=====================================================================
 10   write(*,1000)
      dumfln = ' '
      if (igets(5, dumfln).eq.0) then
        return
      else                                                          1440
        jchar = index(dumfln,' ') - 1
        if (jchar.eq.0) then
          infln = 'tt.dat'
          jchar = 6
        else
          infln = dumfln
        endif
      endif
c
c                                                                   1450
      open(UNIT=2, FILE=infln(1:jchar),
     -             STATUS='OLD',        IOSTAT=ios )
      if (ios.ne.0) then
c -----> CREATE NEW FILE
        open(UNIT=2, FILE=infln(1:jchar), STATUS='NEW',IOSTAT=ios )
        if (ios.ne.0) then
c ---------> CANNOT CREATE NEW FILE
            goto 9990
        endif
```

```
      else                                                              1460
c -----> FILE EXISTS, APPEND NEW DATA
         write(6,1001)
         answer = askno(5,irc)
         if (answer) then
c --------> DO NOT APPEND
            close(UNIT=2)
            goto 10
         endif
      endif
c                                                                       1470
c
c ----> write to file
c
      do 40 k=1,nnn
40       write(UNIT=2, FMT=1020, ERR=9991) xxx(k), yyy(k)
      close(UNIT=2)
      return
c
c ERRORS
9990  call errmsg(' Error opening datafile ')                           1480
      close(UNIT=2)
      goto 10
9991  call errmsg(' Error writing datafile ')
      close(UNIT=2)
      goto 10
1000  format(' >>> Enter filename or <crtl Z> [TT.DAT]: '$)
1001  format(' File already exists! Overwrite ? [Y/N] [N] : '$)
1002  format(a)
1010  format(2i5)
1020  format(2f16.7)                                                    1490
      end
C     LAST UPDATE 16/03/89
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      INTEGER FUNCTION IGETS (ITERM,TBUFF)
```

```
C      IMPLICIT NONE
C
C PURPOSE: READ A CHARACTER STRING FROM THE TERMINAL AND REPORT IF
C          CTRL-Z ISSUED.
C                                                                        1500
C NB : PRINT *,'' is for the CONVEX
C
      INTEGER       ITERM
      CHARACTER*(*) TBUFF
C
C ITERM  : TERMINAL INPUT STREAM
C TBUFF  : TERMINAL BUFFER
C-----------------------------------------------------------------------
      READ  (ITERM,'(A)',END=999) TBUFF
      IF( (ICHAR(TBUFF(1:1)) .EQ. 26) .OR. (ICHAR(TBUFF(1:1)) .EQ. 4) )  1510
     -      GOTO 999
      IGETS=-1
      RETURN
999   IGETS=0
C      PRINT *,''
      RETURN
C
      END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C                                                                        1520
      LOGICAL FUNCTION ASKNO (ITERM,IRC)
C
C Purpose:  Read user response to a yes/no question and either assign
C          the default answer if <cr> is entered or interpret the reply
C          and return the relevant logical value.  If <ctrl-Z> is
C          entered the return code is -1.
C
      INTEGER ITERM,IRC
C
C ITERM  : Terminal input stream                                         1530
C IRC    : Return code 0 - successful
```

```
C                     1 - <ctrl-Z>
C
C Calls   1:  ERRMSG
C
C-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
C Local variables:
C
      INTEGER      JCHAR
      CHARACTER*80 TBUFF                                                  1540
C
C JCHAR  : Nos.  of chars entered
C TBUFF  : Terminal buffer
C
C-------------------------------------------------------------------------
C
      ASKNO=.TRUE.
      IRC=1
C
10    READ (ITERM,'(A)',END=999) TBUFF                                    1550
      JCHAR=INDEX(TBUFF,' ')-1
      IF (JCHAR.EQ.0) THEN
         IRC=0
      ELSEIF (TBUFF(1:1).EQ.'N'.OR.TBUFF(1:1).EQ.'n') THEN
         IRC=0
      ELSEIF (TBUFF(1:1).EQ.'Y'.OR.TBUFF(1:1).EQ.'y') THEN
         IRC=0
         ASKNO=.FALSE.
      ELSE
         CALL ERRMSG ('Error: Reply Y or N, Please re-enter')             1560
         GOTO 10
      ENDIF
999   END
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
      LOGICAL FUNCTION ASKYES (ITERM,IRC)
C
```

```
C Purpose:  Read user response to a yes/no question and either assign
C           the default answer if <cr> is entered or interpret the reply
C           and return the relevant logical value.  If <ctrl-Z> is              1570
C           entered the return code is -1.
C
      INTEGER ITERM,IRC
C
C ITERM  : Terminal input stream
C IRC    : Return code 0 - successful
C                      1 - <ctrl-Z>
C
C Calls   1:  ERRMSG
C                                                                               1580
C-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
C Local variables:
C
      INTEGER      JCHAR
      CHARACTER*80 TBUFF
C
C JCHAR  : Nos.  of chars entered
C TBUFF  : Terminal buffer
C
C-------------------------------------------------------------------------      1590
C
      ASKYES=.TRUE.
      IRC=1
C
10    READ (ITERM,'(A)',END=999) TBUFF
      JCHAR=INDEX(TBUFF,' ')-1
      IF (JCHAR.EQ.0) THEN
         IRC=0
      ELSEIF (TBUFF(1:1).EQ.'N'.OR.TBUFF(1:1).EQ.'n') THEN
         IRC=0                                                                  1600
         ASKYES=.FALSE.
      ELSEIF (TBUFF(1:1).EQ.'Y'.OR.TBUFF(1:1).EQ.'y') THEN
         IRC=0
```

```
      ELSE
         CALL ERRMSG ('Error: Reply Y or N, Please re-enter')
         GOTO 10
      ENDIF
999   END
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C                                                                          1610
      SUBROUTINE ERRMSG (MESAGE)
C
C Purpose:  Print error message in inverse video on a VT100 emulator
C          terminal.
C
      CHARACTER*(*) MESAGE
C
C MESAGE : Error message to be displayed
C
C Calls   0:                                                               1620
C
C-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
C Local variables:
C
      CHARACTER*1 BELL,ESC
C
C BELL    : Bell ascii character
C ESC     : Escape character
C
      DATA BELL/'7'/ , ESC/'27'/                                           1630
C
C---------------------------------------------------------------------------
C
C======REVERSE VIDEO MODE
C
      PRINT *,1000 , ESC , '[' , 7 , 'm'
C
C======PRINT ERROR MESSAGE AND RING BELL
C
```

```
      PRINT *,1010,MESAGE,BELL                                          1640
C
C=====REVERT TO PREVIOUS VIDEO MODE
C
      PRINT *,1020 , ESC , '[' , 0 , 'm'
      RETURN
C
1000  FORMAT (' ',2A1,I1,A1)
1010  FORMAT ('+',A)
1020  FORMAT ('+',2A1,I1,A1)
      END                                                               1650
```

Appendix C

Least-Square Fitting

This is a source code listing of a FORTRAN program which performs the least-square fitting of various models to the experimental data with the option of the incoporation of the resolution function.

```
c     main program
      implicit none
      character*20 filein(3), title, fileout, state, xsectype
      character*50 junk
      real*8 sigq, ave, fave, STP, QBE, Qres
      real*8 reso, ftemp
      integer pnumber, i, estart(3), enumber, imax, j, imequ
      integer iprint, resyn, nfold, sets, ii
      real*8 prm, step, bound, fres, oldchi, sig
      real*8 edat(3,4,200),r1,r2,rl1,rl2,rlam,dlam,rdet                    10
      common/general/estart, enumber(3), prm(200), bound(2,200), imax
      common/resof1/reso(3,200,200),r1,r2,rl1,rl2,rlam,dlam,rdet,resyn
      common/resof2/sig(3,200),nfold
      common/QQQ/ave(200), fave(200), imequ, iprint, sets
c
c
      open(unit=5,file='fitcontrol',status='old')
      rewind(unit=5)
      read(5,1201) title
```

```
      print *,'title :',title                                          20
 3333 read(5,1201) state
      if(state.eq.'stop') go to 4444
      read(5,*) sets
      do 170 i=1,sets
        read(5,1201) filein(i)
        read(5,*) estart(i)
        read(5,*) enumber(i)
 170  continue
 1201 format(a20)
      read(5,1201) fileout                                             30
c
c
      open(unit=10,file='test', status='unknown')
      rewind(unit=10)
      do 190 ii=1,sets
        open(unit=7,file=filein(ii), status='old')
        rewind(unit=7)
        if(estart(ii).eq.1) go to 117
        do 2 i=1,estart(ii)-1
          read(7,1202) junk                                            40
 2      continue
 1202   format(a50)
 117    do 3 i=1, enumber(ii)
          read(7,*) edat(ii,1,i), edat(ii,2,i), edat(ii,3,i)
          if(title.eq.'ORNL') then
            edat(ii,1,i) = edat(ii,1,i)/10.
          endif
          write(10,1023) edat(ii,1,i),edat(ii,2,i),edat(ii,3,i)
 3      continue
        close(unit=7)                                                  50
 190  continue
      close(unit=10)
c
c
      read(5,1201) xsectype
```

```
      print *,xsectype
      read(5,*) pnumber
      do 1 i=1,pnumber
        read(5,*) prm(i), bound(1,i), bound(2,i)
 1    continue                                                    60
c
c  input the predetermined volume fraction as prm(pnumber+1)
c      if(xsectype.eq.'monoshs'.or.xsectype.eq.'polyshs') then
c        read(5,*) prm(pnumber+1)
c        prm(pnumber+1) = prm(pnumber+1)*0.7018
c      endif
c
      read(5,*) step
      read(5,*) ftemp
      read(5,*) imequ                                            70
      read(5,*) imax
      read(5,*) iprint
      read(5,*) resyn
c
c
      if(resyn.eq.0) go to 19
      open(unit=7,file='reso.fun',status='unknown')
      rewind(unit=7)
      nfold = 50
      do 180 ii=1,sets                                           80
        read(5,*) r1
        read(5,*) r2
        read(5,*) rl1
        read(5,*) rl2
        read(5,*) rlam
        read(5,*) dlam
        read(5,*) rdet
        DO 200 I=1,enumber(ii)
          SIG(ii,I)=dSQRT(SIGQ(edat(ii,1,I)))
          STP=6.*SIG(ii,I)/FLOAT(NFOLD)                          90
          QBE=edat(ii,1,I)-0.5*FLOAT(NFOLD)*STP
```

```
          IF(QBE .LE. 0.0) QBE=1.d-10
          DO 300 J=1,nfold
            Qres=QBE+(J-0.5)*STP
            RESO(ii,I,J)=STP*FRES(SIG(ii,I)**2,
     +                  edat(ii,1,I),Qres)
            write(7,1023) Qres,reso(ii,i,j)
 300       continue
 200     continue
 180  continue                                                    100
      close(unit=7)
c
c
 19   call anneal(edat,ftemp,pnumber,step,oldchi,state,xsectype)
c
c
      open(unit=7,file=fileout,status='unknown')
      rewind(unit=7)
      write(7,*) title
      write(7,*) xsectype                                         110
      write(7,1023) oldchi
      write(7,*) ' '
      do 20 i=1,pnumber
        write(7,1023) prm(i),ave(i),fave(i)
 20   continue
 1023 format(3e15.6)
      close(unit=7)
c
      go to 3333
 4444 close(unit=5)                                               120
      stop
      end
c
c
      real*8 function xsec(Q, prm, xsectype)
c
c Scattering Intensity of a Monodispersed Sticky-Hard-Sphere System
```

```
c
      implicit none
      character*20 xsectype                                              130
      real*8 prm(200), Q, eta, a, R, tau, bgd, dsld, pi
      real*8 QR, P, np, mu, alpha
      real*8 Sshs, alp, G1, G2
      real*8 const, thy, d, AA, BB, Z, Z1, Z2, Z3, G11, G12, G13
      real*8 bet, arb, thy1, thy2, f, S, center, deviat, Na
      real*8 L, Vfc, Vhc, fcop, fpfo, foct, rsol, rhc, rfc
      real*8 cylindform, etaa, SHSStr, Vp, phim
      real*8 cylind3form, RectStr, W,liqP,liqSshs
      real*8 H,rhoF,rhoH,rhoS,pstep,sspo,pp(7),liq,phii
      real*8 mstep,step,phi,ssm,fhcfun,ssmn,sspn,ssmo                    140
      real*8 fhcylfun,rv,phio,tt,rhoC,ang
      integer n,j,kk
c
c
        pi = 3.141592653
c
c Monodisperse & Polydisperse Sticky Hard Spheres
c
      if(xsectype.eq.'monoshs'.or.xsectype.eq.'polyshs') then
c Define the parameters                                                  150
c     1. Sticky Sphere Diameter for Structure Factor Calculation
        a = prm(1)
c     2. Hard Sphere Radius for Form Factor Calculation
        R = prm(2)
c     3. Stickiness in Sticky Sphere Model
        tau = prm(3)
c     4. Difference of Scattering Length Density between Spheres and solvent
        dsld = prm(4)
c     5. Incoherent Scattering Background
        bgd = prm(5)                                                     160
        if(xsectype.eq.'monoshs') then
c     6. Volme Fraction
c          eta = prm(6)*a**3./R**3./8.
```

```
        eta = prm(6)
      else
c     6. Width parameter in Schultz distribution
        Z = prm(6)
c     7. Volme Fraction
c       eta = prm(7)*a**3.
c    *        *(Z+1.)**2./(Z+2.)/(Z+3.)/R**3./8.                    170
        eta = prm(7)
      endif
c
c
      if(eta.gt.0.74) then
        xsec=1.d30
        return
      endif
c
c                                                                   180
c Form Factor
      if(xsectype.eq.'monoshs') then
        QR = Q*R
        P = (4.*pi*R**3.*(dsin(QR)-QR*dcos(QR))/QR**3.)**2.
      else
c       alp = (Z+1.)/Q/R
        alp = Q*R/(Z+1.)
        Z1 = Z+1.
        Z2 = Z+2.
        Z3 = Z+3.                                                   190
c       G11 = alp**(-Z1)-(4.+alp**2)**(-Z1/2.)
c    *        *dcos(Z1*datan(2./alp))
c       G12 = Z2*Z1*(alp**(-Z3)+(4.+alp**2)**(-Z3/2)
c    *        *dcos(Z3*datan(2./alp)))
c       G13 = -2.*Z1*(4.+alp**2)**(-Z2/2)
c    *        *dsin(Z2*datan(2./alp))
        G11 = 1-(1.+4.*alp**2.)**(-Z1/2.)*
     *        dcos(Z1*datan(2.*alp))
        G12 = alp**2.*Z1*Z2*(1.+(1.+4.*alp**2.)**(-Z3/2.)
```

```
     *          *dcos(Z3*datan(2.*alp)))                                    200
           G13 = -2.*alp*Z1*(1.+4.*alp**2.)**(-Z2/2.)
     *          *dsin(Z2*datan(2.*alp))
           G1 = G11+G12+G13
c           P = 8*pi**2.*R**6*alp**(Z+7)*G1/Z1**6
           P = 8.*pi**2.*G1/Q**6.
         endif
c
c Structure Factor
         Sshs = SHSStr(Q,eta,a,tau)
c                                                                          210
         if(xsectype.eq.'monoshs') then
           xsec = np*dsld*P*Sshs+bgd
         else
c           G2 = dsin(Z1*datan(1./alp))-Z1*((1+alp**2.)**(-0.5))
c     *          *dcos(Z2*datan(1./alp))
           G2 = dsin(Z1*datan(alp))-Z1*alp*
     *          ((1.+alp**2.)**(-0.5))*dcos(Z2*datan(alp))
           bet = 2.*(1./(1.+alp**2.))**Z1*G2**2./G1
           xsec = np*dsld*P*(1.+bet*(Sshs-1.))+bgd
         endif                                                             220
c
         return
       endif
c
c
       if(xsectype.eq.'polylor') then
c Define the parameters
c      1. Hard Sphere Radius for Form Factor Calculation
         R = prm(1)
c      2. Difference of Scattering Length Density between Spheres and solvent  230
         dsld = prm(2)
c      3. Constant in Lorentzian
         const = prm(3)
c      4. Corelation length in Lorentzian
         thy = prm(4)
```

```
c     5. Incoherent Scattering Background
        bgd = prm(5)
c     6. Width parameter in Schultz distribution
        Z = prm(6)
c                                                                    240
        alp = Q*R/(Z+1.)
        Z1 = Z+1.
        Z2 = Z+2.
        Z3 = Z+3.
        G11 = 1-(1.+4.*alp**2.)**(-Z1/2.)*
     *       dcos(Z1*datan(2.*alp))
        G12 = alp**2.*Z1*Z2*(1.+(1.+4.*alp**2.)**(-Z3/2.)
     *       *dcos(Z3*datan(2.*alp)))
        G13 = -2.*alp*Z1*(1.+4.*alp**2.)**(-Z2/2.)
     *       *dsin(Z2*datan(2.*alp))                                 250
        G1 = G11+G12+G13
        P = 8.*pi**2.*G1/Q**6.
        G2 = dsin(Z1*datan(alp))-Z1*alp*
     *       ((1.+alp**2.)**(-0.5))*dcos(Z2*datan(alp))
        bet = 2.*(1./(1.+alp**2.))**Z1*G2**2./G1
        S = 1.+const/(1.+Q**2*thy**2)
        xsec = dsld*P*(1.+bet*(S-1.))+bgd
        return
      endif
c                                                                    260
c
      if(xsectype.eq.'pshslor') then
c Define the parameters
c     1. Hard Sphere Radius for Form Factor Calculation
        R = prm(1)
c     2. Difference of Scattering Length Density between Spheres and solvent
        dsld = prm(2)
c     3. Constant in Lorentzian
        const = prm(3)
c     4. Corelation length in Lorentzian                             270
        thy = prm(4)
```

```
c     5. Incoherent Scattering Background
        bgd = prm(5)
c     6. Width parameter in Schultz distribution
        Z = prm(6)
c     7. volume fraction in sticky spheres model
        eta = prm(7)
c     8. sticky sphere diameter
        a = prm(8)
c     9. stickness                                                    280
        tau = prm(9)
c     10. fraction of sticky spheres part in structrure factor
        f = prm(10)
c
        np = 6.*eta/pi/a**3.
        alp = Q*R/(Z+1.)
        Z1 = Z+1.
        Z2 = Z+2.
        Z3 = Z+3.
        G11 = 1-(1.+4.*alp**2.)**(-Z1/2.)*                          290
     *      dcos(Z1*datan(2.*alp))
        G12 = alp**2.*Z1*Z2*(1.+(1.+4.*alp**2.)**(-Z3/2.)
     *      *dcos(Z3*datan(2.*alp)))
        G13 = -2.*alp*Z1*(1.+4.*alp**2.)**(-Z2/2.)
     *      *dsin(Z2*datan(2.*alp))
        G1 = G11+G12+G13
        P = 8.*pi**2.*G1/Q**6.
        G2 = dsin(Z1*datan(alp))-Z1*alp*
     *      ((1.+alp**2.)**(-0.5))*dcos(Z2*datan(alp))
        bet = 2.*(1./(1.+alp**2.))**Z1*G2**2./G1                    300
c Structure Factor
        Sshs = SHSStr(Q,eta,a,tau)
        S = f*Sshs+(1.-f)*(1.+const/(1.+Q**2*thy**2))
c
        xsec = dsld*P*(1.+bet*(S-1.))+bgd
        return
      endif
```

```
c
c
      if(xsectype.eq.'mshslor') then                                    310
c Define the parameters
c     1. Hard Sphere Radius for Form Factor Calculation
        R = prm(1)
c     2. Difference of Scattering Length Density between Spheres and solvent
        dsld = prm(2)
c     3. Constant in Lorentzian
        const = prm(3)
c     4. Corelation length in Lorentzian
        thy = prm(4)
c     5. Incoherent Scattering Background                               320
        bgd = prm(5)
c     6. volume fraction in sticky spheres model
        eta = prm(6)
c     7. sticky sphere diameter
        a = prm(7)
c     8. stickness
        tau = prm(8)
c     9. fraction of sticky spheres part in structrure factor
        f = prm(9)
c                                                                       330
        np = 6.*eta/pi/a**3.
c Form Factor
        QR = Q*R
        P = (4.*pi*R**3.*(dsin(QR)-QR*dcos(QR))/QR**3.)**2.
c Structure Factor
        Sshs = SHSStr(Q,eta,a,tau)
c
       xsec = f*np*dsld*P*Sshs+
     +        (1.-f)*(const/(1.+Q**2*thy**2))+bgd
        return                                                          340
       endif
c
c
```

```
      if(xsectype.eq.'mcylind') then
c Define the parameters
c     1. Radius of the cylinder
        R = prm(1)
c     2. Length of the cylinder
        L = prm(2)
c     3. Core scattering length density                                    350
        rhoC = prm(3)
c     4. Solvent scattering length density
        rhoS = prm(4)
c     5. Particle volume fraction
        eta = prm(5)
c     6. Background
        bgd = prm(6)
c
      P = cylindform(Q,R,L)
      xsec = 1.d8*eta*pi*R**2.d0*L*(rhoC-rhoS)**2.d0*P+bgd                 360
      return
      endif
c
c
      if(xsectype.eq.'mcylshs') then
c Define the parameters
c     1. Aggregation Number
        Na = prm(1)
c     2. volume fraction in sticky spheres model
        eta = prm(2)                                                       370
c     3  Adjustment to the absolute scale
        const = prm(3)
c     4. Incoherent Scattering Background (1/cm)
        bgd = prm(4)
c     5. stickness
        tau = prm(5)
c     6. Length of Cylinder in Form Factor Calculation (A)
        L = prm(6)
c     7. Length of the perfluorocarbon chain in copolymer (A)
```

```
      F = prm(7)                                                        380
c     8. Volume of fluorocarbon chain in copolymer (A^3)
      Vfc = prm(8)
c     9. Volume of hydrocarbon chain in copolymer (A^3)
      Vhc = prm(9)
c     10. Copolymer volume fraction
      fcop = prm(10)
c     11. Perflrorooctane volume fraction
      fpfo = prm(11)
c     12. iso-octane volume fraction
      foct = prm(12)                                                    390
c     13. Scattering length density of fluorocarbon (A^-2)
      rfc = prm(13)
c     14. Scattering length density of hydrocarbon (A^-2)
      rhc = prm(14)
c     15. Cylinder Cross Section Radius for Form Factor Calculation
      R = prm(15)
c     16. sticky sphere diameter
      a = prm(16)
c     17. correlation length in Lorentzian
      thy = prm(17)                                                     400
c     18. constant for Lorentzian
      arb = prm(18)
c
c
c     first approximation 5/27/94
      R = dsqrt(Na*Vhc/21.74d0/pi)
c      a = (6.*R**2.*(L+2.*F))**(1./3.)
      etaa = 6.*eta*R**2.*L/a**3.
      rsol = (rhc*(fcop*Vhc/(Vhc+Vfc)+foct-etaa)+
     +        rfc*(fcop*Vfc/(Vhc+Vfc)+fpfo))/(1-etaa)                   410
c Form Factor
      P = cylindform(Q,R,L)
c Structure Factor
      Sshs = SHSStr(Q,eta,a,tau)
c
```

```
      xsec = 6.d8*const*eta*Na**2.*
     *       (L/21.74)**2.*(3.635d-3-Vhc*rsol)**2.*
     *       P*Sshs/a**3./pi+arb/(1.+thy**2.*Q**2.)+bgd
      return
      endif                                                          420
c
c
      if(xsectype.eq.'m3cyshs') then
c Define the parameters
c     1. Aggregation Number
        Na = prm(1)
c     2. volume fraction in sticky spheres model
        eta = prm(2)
c     3 Adjustment to the absolute scale
        const = prm(3)                                               430
c     4. Incoherent Scattering Background (1/cm)
        bgd = prm(4)
c     5. stickness
        tau = prm(5)
c     6. Length of the hydrocarbon chain in copolymer (A)
        L = prm(6)
c     7. Length of the perfluorocarbon chain in copolymer (A)
        F = prm(7)
c     8. Volume of fluorocarbon chain in copolymer (A^3)
        Vfc = prm(8)                                                 440
c     9. Volume of hydrocarbon chain in copolymer (A^3)
        Vhc = prm(9)
c     10. Copolymer volume fraction
        fcop = prm(10)
c     11. Perflrorooctane volume fraction
        fpfo = prm(11)
c     12. iso-octane volume fraction
        foct = prm(12)
c     13. Scattering length density of fluorocarbon (A^-2)
        rfc = prm(13)                                                450
c     14. Scattering length density of hydrocarbon (A^-2)
```

```
      rhc = prm(14)
c     15. Cylinder Cross Section Radius for Form Factor Calculation
      R = prm(15)
c     16. sticky sphere diameter
      a = prm(16)
c     17. correlation length in Lorentzian
      thy = prm(17)
c     18. constant for Lorentzian
      arb = prm(18)                                                    460
c     19. solvent scattering length density correction factor
      alpha = prm(19)
c
c
c     first approximation 5/27/94
      np = 6.d0*eta/pi/a**3.d0
      Vp = pi*R**2.d0*(L+2.d0*F)
      phim = np*Vp
      rsol = alpha*(rhc*((Vhc/(Vfc+Vhc))*(fcop-phim)+foct)+
     +       rfc*((Vfc/(Vfc+Vhc))*(fcop-phim)+fpfo))/(1-phim)          470
c Form Factor
      P = cylind3form(Q,R,L,F,rfc,rhc,rsol)
c Structure Factor
      Sshs = SHSStr(Q,eta,a,tau)
c
      xsec = 6.d8*const*eta*
     *       (pi*R**2.d0*(L+2.d0*F))**2.d0*
     *       P*Sshs/a**3./pi+arb/(1.+thy**2.*Q**2.)+bgd
      return
     endif                                                             480
c
c
     if(xsectype.eq.'complex') then
c Define the parameters
c     1. volume fraction of crystal
      phii = prm(1)
c     2. Adjustment to the absolute scale
```

```
      const = prm(2)
c     3. Incoherent Scattering Background (1/cm)
      bgd = prm(3)                                                    490
c     4. stickness
      tau = prm(4)
c     5. Length of the hydrocarbon chain in copolymer (A)
      H = prm(5)
c     6. Length of the perfluorocarbon chain in copolymer (A)
      F = prm(6)
c     7. Scattering length density of fluorocarbon (A^-2)
      rhoF = prm(7)
c     8. Scattering length density of hydrocarbon (A^-2)
      rhoH = prm(8)                                                   500
c     9. Cylinder Cross Section Radius for liquid Form Factor Calculation
      R = prm(9)
c     10. sticky sphere diameter
      a = prm(10)
c     11. Volume fraction of micelles
      eta = prm(11)*a**3.d0/R**2.d0/(2.d0*F+H)/6.d0
c     12. Number of lamellar layers in a crystal
      n = prm(12)
c     13. Length of the crystal
      L = prm(13)                                                     510
c     14. Width of the crystal
      W = prm(14)
c     15. Steps size for the integration in crystal part
      step = prm(15)
c     16. Crystal part on/off switch (0. is off)
      arb = prm(16)
c     17. Volume ratio of PFO to i-OCT
      rv = prm(17)
c     18. Total volume fraction of COP in the sample
      phio = prm(18)                                                  520
c     19. Copolymer angle
      ang = prm(19)
c
```

```
c
c     date 11/4/95
c
c Liquid Form Factor
      if(rv.eq.0.81d0) then
        rhoS = 1.028d-5
      elseif(rv.eq.2.9d0) then                                    530
        tt = 2.d0*F+H
        rhoS = ((phio-phii-prm(11))
     +          *1.028d-5+(1.d0-phio)*1.238d-5)
     +          /(1.d0-phii-prm(11))
      else
        print *, 'Wrong Volume Ratio'
        return
      endif
      liqP = cylind3form(Q,R,H,F,rhoF,rhoH,rhoS)
c Liquid Structure Factor                                         540
      liqSshs = SHSStr(Q,eta,a,tau)
c
      liq = 6.d8*(1.d0-phii)*eta*
     *      (pi*R**2.d0*(H+2.d0*F))**2.d0*
     *      liqP*liqSshs/a**3./pi
c Crystal part
      H = H*ang
      F = F*ang
      ssm = 0.d0
      if(arb.eq.0.d0) go to 407                                   550
      if(rv.eq.0.81d0) then
        rhoS = 1.028d-5
      elseif(rv.eq.2.9d0) then
        rhoS = ((phio-phii)*1.028d-5+(1.d0-phio)*1.238d-5)
     +          /(1.d0-phii)
      else
        print *, 'Wrong Volume Ratio'
        return
      endif
```

```
      mstep = 1.d0/step                                                  560
      pstep = pi/2.d0/step
      ssmo = 0.d0
      sspo = fhcfun(0.d0,0.d0,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
      do 405 kk = 1,step
        phi = dfloat(kk)*pstep
        sspn = fhcfun(0.d0,phi,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
        ssmo = ssmo+pstep*(sspo+sspn)/2.d0
        sspo = sspn
405    continue
      do 205 j = 1,step                                                  570
        mu = dfloat(j)*mstep
        sspo = fhcfun(mu,0.d0,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
        ssmn = 0.d0
        do 305 kk = 1,step
          phi = dfloat(kk)*pstep
          sspn = fhcfun(mu,phi,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
          ssmn = ssmn+pstep*(sspo+sspn)/2.d0
          sspo = sspn
305      continue
        ssm = ssm+mstep*(ssmo+ssmn)/2.d0                                 580
        ssmo = ssmn
205    continue
c
407    xsec = const*(2.d8*ssm*phii*W*L*n*(2.d0*F+H)/pi+liq)+bgd
c
      return
    endif
c
c
    if(xsectype.eq.'rect3cy') then                                       590
c Define the parameters
c     1. volume fraction in sticky spheres model
      eta = prm(1)
c     2. Adjustment to the absolute scale
      const = prm(2)
```

```
c     3. Incoherent Scattering Background (1/cm)
        bgd = prm(3)
c     4. stickness
        tau = prm(4)
c     5. Length of the hydrocarbon chain in copolymer (A)                    600
        H = prm(5)
c     6. Length of the perfluorocarbon chain in copolymer (A)
        F = prm(6)
c     7. Scattering length density of the solvent
        rhoS = prm(7)
c     8. Scattering length density of fluorocarbon (A^-2)
        rhoF = prm(8)
c     9. Scattering length density of hydrocarbon (A^-2)
        rhoH = prm(9)
c     10. Cylinder Cross Section Radius for Form Factor Calculation          610
        R = prm(10)
c     11. sticky sphere diameter
        a = prm(11)
c     12. half with of the rectangular distribution
        W = prm(12)
c
c
c date      11/4/95
        if(W.gt.a) W = a
c Form Factor                                                              620
        P = cylind3form(Q,R,H,F,rhoF,rhoH,rhoS)
c Structure Factor
        Sshs = RectStr(Q,eta,a,W)
c
        xsec = 6.d8*const*eta*
     *          (pi*R**2.d0*(L+2.d0*F))**2.d0*
     *          P*Sshs/a**3./pi+bgd
        return
      endif
c                                                                          630
c
```

```
      if(xsectype.eq.'lorentzian') then
        const = prm(1)
        thy = prm(2)
        bgd = prm(3)
c
        xsec = const/(1.+Q**2*thy**2)+bgd
c
        return
      endif                                                          640
c
c
      if(xsectype.eq.'teubnerstrey') then
        d = prm(1)
        thy = prm(2)
        dsld = prm(3)
        bgd = prm(4)
c
        AA = (1./thy**2.+(2.*pi/d)**2.)**2.
        BB = 2.*(1./thy**2.-(2.*pi/d)**2.)                           650
c
        xsec = 8.*pi*dsld/(AA+BB*Q**2.+Q**4.)/thy+bgd
c
        return
      endif
c
c
      if(xsectype.eq.'teubadd') then
        d = prm(1)
        thy1 = prm(2)                                                660
        thy2 = prm(3)
        const = prm(4)
        arb = prm(5)
        bgd = prm(6)
c
        AA = (1./thy1**2.+(2.*pi/d)**2.)**2.
        BB = 2.*(1./thy1**2.-(2.*pi/d)**2.)
```

```
c
      xsec = const/(AA+BB*Q**2.+Q**4.)
     +       +const*arb/(1+Q**2.*thy2**2.)**2.+bgd                    670
c
      return
     endif
c
c
     if(xsectype.eq.'gaulor1') then
      center = prm(1)
      deviat = prm(2)
      const = prm(3)
      thy = prm(4)                                                    680
      arb = prm(5)
      bgd = prm(6)
c
      xsec = const*exp(-(Q-center)**2./deviat**2./2.)
     +       +arb/(1.+thy**2.*Q**2.)+bgd
c
      return
     endif
c
c                                                                     690
     if(xsectype.eq.'fhcryst') then
      eta = prm(1)
      n = prm(2)
      F = prm(3)
      H = prm(4)
      L = prm(5)
      W = prm(6)
      rhoF = prm(7)
      rhoH = prm(8)
      rhoS = prm(9)                                                   700
      const = prm(10)
      bgd = prm(11)
      step = prm(12)
```

```
c       R = prm(13)
        pp(1) = -5.4325d4
        pp(2) = 5.6746d4
        pp(3) = -2.1513d4
        pp(4) = 3.4672d3
        pp(5) = -2.1285d2
        pp(6) = 5.5331d0                                                    710
        pp(7) = 3.8840d-1
        mstep = 1.d0/step
        pstep = pi/2.d0/step
        ssmo = 0.d0
        sspo = fhcfun(0.d0,0.d0,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
        do 400 kk = 1,step
          phi = dfloat(kk)*pstep
          sspn = fhcfun(0.d0,phi,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
          ssmo = ssmo+pstep*(sspo+sspn)/2.d0
          sspo = sspn                                                       720
 400    continue
        ssm = 0.d0
        do 200 j = 1,step
          mu = dfloat(j)*mstep
          sspo = fhcfun(mu,0.d0,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
          ssmn = 0.d0
          do 300 kk = 1,step
            phi = dfloat(kk)*pstep
            sspn = fhcfun(mu,phi,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
            ssmn = ssmn+pstep*(sspo+sspn)/2.d0                              730
            sspo = sspn
 300      continue
          ssm = ssm+mstep*(ssmo+ssmn)/2.d0
          ssmo = ssmn
 200    continue
        liq = pp(1)*Q**6.d0+pp(2)*Q**5.d0+pp(3)*Q**4.d0
     +       +pp(4)*Q**3.d0+pp(5)*Q**2.d0+pp(6)*Q+pp(7)
c
c       S = 1.d0-(3*(dsin(Q*R)-Q*R*dcos(Q*R))/(Q*R)**3.d0)
```

```
c                                                                      740
c       xsec = 2.d8*S*ssm*eta*W*L*n*(2.d0*F+H)/pi+const*liq+bgd
        xsec = const*(2.d8*ssm*eta*W*L*n*(2.d0*F+H)/pi
     +         +(1.d0-eta)*liq)+bgd
c
        return
      endif
c
c
      if(xsectype.eq.'fhcylin') then
        eta = prm(1)                                                   750
        n = prm(2)
        F = prm(3)
        H = prm(4)
        L = prm(5)
        rhoF = prm(6)
        rhoH = prm(7)
        rhoS = prm(8)
        const = prm(9)
        bgd = prm(10)
        step = prm(11)                                                 760
        R = n*(2.d0*F+H)+F+H
        pp(1) = -5.4325d4
        pp(2) = 5.6746d4
        pp(3) = -2.1513d4
        pp(4) = 3.4672d3
        pp(5) = -2.1285d2
        pp(6) = 5.5331d0
        pp(7) = 3.8840d-1
        mstep = 1.d0/step
        ssmo = fhcylfun(0.d0,L,F,H,n,rhoF,rhoH,rhoS,Q)                 770
        ssm = 0.d0
        do 209 j = 1,step
          mu = dfloat(j)*mstep
          ssmn = fhcylfun(mu,L,F,H,n,rhoF,rhoH,rhoS,Q)
          ssm = ssm+mstep*(ssmo+ssmn)/2.d0
```

```
          ssmo = ssmn
 209      continue
        liq = pp(1)*Q**6.d0+pp(2)*Q**5.d0+pp(3)*Q**4.d0
     +        +pp(4)*Q**3.d0+pp(5)*Q**2.d0+pp(6)*Q+pp(7)
c                                                                    780
        xsec = const*(1.d8*ssm*eta*R**2.d0*L*pi
     +          +(1.d0-eta)*liq)+bgd
c
        return
      endif
c
      print *,'Wrong Xsec Type!'
      xsec = 1.d30
      return
      end                                                            790
c
c
      real*8 function fhcylfun(mu,L,F,H,n,rhoF,rhoH,rhoS,Q)
      implicit none
      real*8 mu,L,F,H,rhoF,rhoH,rhoS,Q,t,Qz,Qj1tt,Qj1t,Qj2tt
      real*8 R,F1,F2,QR,QRt,pi,Qj2t,bessj1
      integer n,j
      pi = 3.141592653
      t = 2.d0*F+H
      R = n*t+F+H                                                    800
      F1 = 0.d0
      if(mu.eq.0.d0) then
        Qj2tt = Q*H
        Qj2t = 2.d0*bessj1(Qj2tt)/Qj2tt
        F2 = (H/R)**2.d0*Qj2t
        Qz = 1.d0
        QRt = Q*R
        QR = 2.d0*bessj1(QRt)/QRt
        if(n.gt.0) then
         do 10 j=1,n                                                 810
          Qj1tt = Q*dfloat(j)*t
```

```
      Qj1t = (dfloat(j)*t/R)**2.d0*2.d0*bessj1(Qj1tt)/Qj1tt
      Qj2tt = Q*(dfloat(j)*t+H)
      Qj2t = ((dfloat(j)*t+H)/R)**2.d0*2.d0*bessj1(Qj2tt)/Qj2tt
      F1 = F1+(Qj1t-Qj2t)
10    continue
    endif
  elseif(mu.eq.1.d0) then
    F2 = (H/R)**2.d0
    Qz = sin(Q*L/2.d0)/(Q*L/2.d0)                                    820
    QR = 1.d0
    if(n.gt.0) then
      do 20 j=1,n
        Qj1t = (dfloat(j)*t/R)**2.d0
        Qj2t = ((dfloat(j)*t+H)/R)**2.d0
        F1 = F1+(Qj1t-Qj2t)
20    continue
    endif
  else
    Qj2tt = Q*H*dsqrt(1.d0-mu**2.d0)                                 830
    Qj2t = 2.d0*bessj1(Qj2tt)/Qj2tt
    F2 = (H/R)**2.d0*Qj2t
      Qz = dsin(Q*mu*L/2.d0)/(Q*mu*L/2.d0)
      QRt = Q*R*dsqrt(1.d0-mu**2.d0)
      QR = 2.d0*bessj1(QRt)/QRt
    if(n.gt.0) then
     do 30 j=1,n
      Qj1tt = Q*t*dfloat(j)*dsqrt(1.d0-mu**2.d0)
      Qj1t = (dfloat(j)*t/R)**2.d0*2.d0*bessj1(Qj1tt)/Qj1tt
      Qj2tt = Q*(t*dfloat(j)+H)*dsqrt(1.d0-mu**2.d0)                 840
      Qj2t = ((dfloat(j)*t+H)/R)**2.d0*2.d0*bessj1(Qj2tt)/Qj2tt
      F1 = F1+(Qj1t-Qj2t)
30    continue
    endif
  endif
  fhcylfun = (Qz*((rhoF-rhoH)*(F1-F2)+(rhoF-rhoS)*QR))**2.d0
  return
```

```
      end
c
c                                                                       850
      real*8 function fhcfun(mu,phi,L,W,F,H,n,rhoF,rhoH,rhoS,Q)
      implicit none
      real*8 mu,phi,L,W,F,H,rhoF,rhoH,rhoS,Q,t,QzF2,QzF,QzH,Qx,Qy
      real*8 F1,F2,F3,F4,F6,F7,Qxx,Qyy,pi
      integer m,n,j
      pi = 3.141592653
      t = 2.d0*F+H
      m = n/2
      F6 = 0.d0
      F7 = 0.d0                                                         860
      do 10 j=1,m
        F6=F6+dcos(Q*mu*dfloat(n-2*j)*t/2.d0)
        F7=F7+dcos(Q*mu*dfloat(n+1-2*j)*t/2.d0)
10    continue
      if(mu.eq.0.d0) then
        QzF2 = 1.d0
        QzF = 1.d0
        QzH = 1.d0
      else
        QzF2 = dsin(Q*mu*F/2.d0)/(Q*mu*F/2.d0)                          870
        QzF = dsin(Q*mu*F)/(Q*mu*F)
        QzH = dsin(Q*mu*H/2.d0)/(Q*mu*H/2.d0)
      endif
      if(mu.eq.1.d0) then
        Qxx = 1.d0
        Qyy = 1.d0
      elseif(phi.eq.0.d0) then
        Qx = Q*dsqrt(1.d0-mu**2.d0)*L/2.d0
        Qxx = dsin(Qx)/Qx
        Qyy = 1.d0                                                      880
      elseif(phi.eq.(pi/2.d0)) then
        Qxx = 1.d0
        Qy = Q*dsqrt(1.d0-mu**2.d0)*W/2.d0
```

```
      Qyy = dsin(Qy)/Qy
    else
      Qx = Q*dsqrt(1.d0-mu**2.d0)*dcos(phi)*L/2.d0
      Qxx = dsin(Qx)/Qx
      Qy = Q*dsqrt(1.d0-mu**2.d0)*dsin(phi)*W/2.d0
      Qyy = dsin(Qy)/Qy
    endif                                                          890
    F1 = Qxx*Qyy/dfloat(n)/t
    F2 = 2.d0*F*(rhoF-rhoS)*QzF2*dcos(Q*mu*(dfloat(n)*t-F)/2.d0)
    F3 = 4.d0*F*(rhoF-rhoS)*QzF*(F6+dfloat(n-1)/2.d0-dfloat(m))
    F4 = 2.d0*H*(rhoH-rhoS)*QzH*(dfloat(n)/2.d0-dfloat(m)+F7)
    fhcfun = (F1*(F2+F3+F4))**2.d0
    return
    end
c
c
    real*8 function SHSStr(Q, eta, a, tau)                         900
c
c This function is to calculate the structure factor of sticky hard spheres
c
c
    implicit none
    real*8 Q,eta,a,tau,K,gamma,epsilon,lambda,mu,alpha,beta
    real*8 c1,c2,c3,c4,c5,c
c
      K = Q*a
      gamma = eta*(1.+eta/2.)/(1.-eta)**2./3.                      910
      epsilon = tau+eta/(1.-eta)
      if((epsilon**2.-gamma).lt.0.) then
        SHSStr = 1.d30
c       print *,'Negative Root!'
        return
      endif
      lambda = 6.*(epsilon-dsqrt(epsilon**2.-gamma))/eta
      mu = lambda*eta*(1.-eta)
      alpha = (1.+2.*eta-mu)**2./(1.-eta)**4.
```

```
      beta = -(3.*eta*(2.+eta)**2.-2.*mu*(1.+7.*eta+eta**2.)+                920
     +        mu**2.*(2.+eta))/(1.-eta)**4./2.
      c1 = alpha*K**3.*(dsin(K)-K*dcos(K))
      c2 = beta*K**2.*(2.*K*dsin(K)-(K**2.-2.)*dcos(K)-2.)
      c3 = eta*alpha*((4.*K**3-24.*K)*dsin(K)-
     +       (K**4.-12.*K**2.+24.)*dcos(K)+24.)/2.
      c4 = -2.*eta**2.*lambda**2.*(1.-dcos(K))/K**2
      c5 = 2.*eta*lambda*sin(K)/K
      c = -24.*eta*(c1+c2+c3)/K**6.+c4+c5
      SHSStr = 1./(1.-c)
c                                                                            930
      return
      end
c
c
      real*8 function HSStr(Q, eta, a)
c
c This function is to calculate the structure factor of hard spheres
c
c
      implicit none                                                          940
      real*8 Q,eta,a,K,gamma,alpha,beta,pi,np
      real*8 c1,c2,c3,c
c
      pi = 3.141592653
c
      K = Q*a
      np = 6.d0*eta/pi/a**3.d0
      alpha = (1.d0+2.d0*eta)**2.d0/(1.d0-eta)**4.d0
      beta = -6.d0*eta*(1.d0+eta/2.d0)**2.d0/(1.d0-eta)**4.d0
      gamma = eta*alpha/2.d0                                                 950
      c1 = alpha*K**3.d0*(dsin(K)-K*dcos(K))
      c2 = beta*K**2.d0*(2.d0*K*dsin(K)-(K**2.d0-2.d0)*dcos(K)-2.d0)
      c3 = gamma*((4.d0*K**3.d0-24.d0*K)*dsin(K)-
     -     (K**4.d0-12.d0*K**2.d0+24.d0)*dcos(K)+24.d0)
      c = -4*pi*a**3.d0*(c1+c2+c3)/K**6.d0
```

```
c
      HSStr = 1.d0/(1.d0-np*c)
c
      return
      end                                                          960
c
c
      real*8 function RectStr(Q, eta, a, W)
c
c This function is to calculate the structure factor of a rectangularly
c distributed polydispersed hard-sphere system
c
      implicit none
      real*8 Q,eta,a,beta
      real*8 c1,c2,c3,c4,c5,c,W,HSStr,S,QW,QR                      970
c
      S = HSStr(Q,eta,a)
c
      QW = Q*W
      QR = Q*a/2.d0
c
      c1 = -QW/2.d0+QW*QR**2.d0+QW**3.d0/3.d0-
     -     5.d0*dcos(2.d0*QR)*dsin(QW)*dcos(QW)/2.d0
      c2 = (QR**2.d0+QW**2.d0)*dcos(2.d0*QR)*dsin(2.d0*QW)/2.d0
      c3 = QR*QW*dsin(2.d0*QR)*dcos(2.d0*QW)+                      980
     +     3.d0*QW*(dcos(QR)*dcos(QW))**2.d0
      c4 = 3.d0*QW*(dsin(QR)*dsin(QW))**2.d0-
     -     6.d0*QR*dcos(QR)*dsin(QR)*dcos(QW)*dsin(QW)
      c5 = c1+c2+c3+c4
      c = 2.d0*dsin(QR)*dsin(QW)-QR*dcos(QR)*dsin(QW)-
     -    QW*dsin(QR)*dcos(QW)

      beta = 2.d0*c**2.d0/c5/QW
c
      RectStr = 1.d0+beta*(S-1.d0)                                 990
c
```

```
      return
      end
c
c
      real*8 function cylindform(Q, R, L)
c
c This function is to calculate the form factor of cylinder
c
c                                                                    1000
      implicit none
      real*8 Q,R,L,QR,QL,CFfunction,P1,P2,PP,QQ,RR,LL,bessj1
      common/CFF/QQ,RR,LL
      QQ = Q
      RR = R
      LL = L
      QR = Q*R
      QL = Q*L/2
      P1 = ((2.*bessj1(QR)/QR)**2.+
     +      CFfunction(1.d-4))*0.0001d0/2.                         1010
      P2 = (CFfunction(0.9999d0)+(dsin(QL)/QL)**2.)*0.0001d0/2.
      call qromb(0.0001d0,0.9999d0,PP)
      cylindform = P1+PP+P2
      return
      end
c
c
      real*8 function cylind3form(Q, R, L, t,rhoo,rhoi,rhos)
c
c This function is to calculate the form factor of a 3-layer cylinder  1020
c
c
      implicit none
      real*8 Q,R,L,QR,QL,CF3function,P1,P2,PP,QQ,RR,LL,bessj1
      real*8 t,tt,rhoo,rhooo,rhoi,rhoii,rhos,rhoss,QLt,PP1,PP2
      common/CFF3/QQ,RR,LL,tt,rhooo,rhoii,rhoss
      QQ = Q
```

```
      RR = R
      LL = L
      tt = t                                                          1030
      rhooo = rhoo
      rhoii = rhoi
      rhoss = rhos
      QR = Q*R
      QLt = Q*(L+2.d0*t)/2
      QL = Q*L/2
      PP1 = ((rhoo-rhos)-L*(rhoo-rhoi)/(L+2.d0*t))*
     *     2*bessj1(QR)/QR
      P1 = (PP1**2.d0+CF3function(1.d-4))*1.d-4/2.d0
      PP2 = (rhoo-rhos)*dsin(QLt)/QLt-                                1040
     -     (L/(L+2.d0*t))*(rhoo-rhoi)*dsin(QL)/QL
      P2 = (CF3function(0.9999d0)+PP2**2.d0)*0.0001d0/2.d0
      call qromb3(0.0001d0,0.9999d0,PP)
      cylind3form = P1+PP+P2
      return
      end
c
c
      real*8 FUNCTION bessj1(x)
      REAL*8 x                                                        1050
      REAL*8 ax,xx,z
      DOUBLE PRECISION p1,p2,p3,p4,p5,q1,q2,q3,q4,q5,r1,r2,r3,r4,r5,r6,
     *s1,s2,s3,s4,s5,s6,y
      SAVE p1,p2,p3,p4,p5,q1,q2,q3,q4,q5,r1,r2,r3,r4,r5,r6,s1,s2,s3,s4,
     *s5,s6
      DATA r1,r2,r3,r4,r5,r6/72362614232.d0,-7895059235.d0,
     *242396853.1d0,-2972611.439d0,15704.48260d0,-30.16036606d0/,s1,s2,
     *s3,s4,s5,s6/144725228442.d0,2300535178.d0,18583304.74d0,
     *99447.43394d0,376.9991397d0,1.d0/
      DATA p1,p2,p3,p4,p5/1.d0,.183105d-2,-.3516396496d-4,            1060
     *.2457520174d-5,-.240337019d-6/, q1,q2.q3,q4,q5/.04687499995d0,
     *-.2002690873d-3,.8449199096d-5,-.88228987d-6,.105787412d-6/
      if(abs(x).lt.8.)then
```

```
      y=x**2
      bessj1=x*(r1+y*(r2+y*(r3+y*(r4+y*(r5+y*r6))))/(s1+y*(s2+y*(s3+
     *y*(s4+y*(s5+y*s6)))))
      else
        ax=abs(x)
        z=8./ax
        y=z**2                                                    1070
        xx=ax-2.356194491
        bessj1=dsqrt(.636619772/ax)*(cos(xx)*(p1+y*(p2+y*(p3+y*(p4+y*
     *p5))))-z*sin(xx)*(q1+y*(q2+y*(q3+y*(q4+y*q5)))))
      endif
      return
      END
c
c
      SUBROUTINE qromb(a,b,ss)
      INTEGER JMAX,JMAXP,K,KM                                     1080
      REAL*8 a,b,ss,EPS
      PARAMETER (EPS=1.e-6, JMAX=20, JMAXP=JMAX+1, K=5, KM=K-1)
CU    USES polint,trapzd
      INTEGER j
      REAL*8 dss,h(JMAXP),s(JMAXP)
      h(1)=1.
      do 11 j=1,JMAX
        call trapzd(a,b,s(j),j)
        if (j.ge.K) then
          call polint(h(j-KM),s(j-KM),K,0.,ss,dss)                1090
          if (abs(dss).le.EPS*abs(ss)) return
        endif
        s(j+1)=s(j)
        h(j+1)=0.25*h(j)
11    continue
      pause 'too many steps in qromb'
      END
c
c
```

```
      SUBROUTINE qromb3(a,b,ss)                                          1100
      INTEGER JMAX,JMAXP,K,KM
      REAL*8 a,b,ss,EPS
      PARAMETER (EPS=1.e-6, JMAX=20, JMAXP=JMAX+1, K=5, KM=K-1)
CU    USES polint,trapzd
      INTEGER j
      REAL*8 dss,h(JMAXP),s(JMAXP)
      h(1)=1.
      do 11 j=1,JMAX
        call trapzd3(a,b,s(j),j)
        if (j.ge.K) then                                                 1110
          call polint(h(j-KM),s(j-KM),K,0.,ss,dss)
          if (abs(dss).le.EPS*abs(ss)) return
        endif
        s(j+1)=s(j)
        h(j+1)=0.25*h(j)
11    continue
      pause 'too many steps in qromb'
      END
c
c                                                                        1120
      SUBROUTINE trapzd(a,b,s,n)
      INTEGER n
      REAL*8 a,b,s,CFfunction
      external CFfunction
      INTEGER it,j
      REAL*8 del,sum,tnm,x
      if (n.eq.1) then
        s=0.5*(b-a)*(CFfunction(a)+CFfunction(b))
      else
        it=2**(n-2)                                                      1130
        tnm=it
        del=(b-a)/tnm
        x=a+0.5*del
        sum=0.
        do 11 j=1,it
```

```
        sum=sum+CFfunction(x)
        x=x+del
11      continue
       s=0.5*(s+(b-a)*sum/tnm)
      endif                                                    1140
      return
      END
c
c
      SUBROUTINE trapzd3(a,b,s,n)
      INTEGER n
      REAL*8 a,b,s,CF3function
      external CF3function
      INTEGER it,j
      REAL*8 del,sum,tnm,x                                     1150
      if (n.eq.1) then
       s=0.5*(b-a)*(CF3function(a)+CF3function(b))
      else
       it=2**(n-2)
       tnm=it
       del=(b-a)/tnm
       x=a+0.5*del
       sum=0.
       do 11 j=1,it
        sum=sum+CF3function(x)                                 1160
        x=x+del
11      continue
       s=0.5*(s+(b-a)*sum/tnm)
      endif
      return
      END
c
c
      SUBROUTINE polint(xa,ya,n,x,y,dy)
      INTEGER n,NMAX                                           1170
      REAL*8 dy,x,y,xa(n),ya(n)
```

```
      PARAMETER (NMAX=10)
      INTEGER i,m,ns
      REAL*8 den,dif,dift,ho,hp,w,c(NMAX),d(NMAX)
      ns=1
      dif=abs(x-xa(1))
      do 11 i=1,n
        dift=abs(x-xa(i))
        if (dift.lt.dif) then
          ns=i                                                    1180
          dif=dift
        endif
        c(i)=ya(i)
        d(i)=ya(i)
11    continue
      y=ya(ns)
      ns=ns-1
      do 13 m=1,n-1
        do 12 i=1,n-m
          ho=xa(i)-x                                              1190
          hp=xa(i+m)-x
          w=c(i+1)-d(i)
          den=ho-hp
          if(den.eq.0.)pause 'failure in polint'
          den=w/den
          d(i)=hp*den
          c(i)=ho*den
12      continue
        if (2*ns.lt.n-m)then
          dy=c(ns+1)                                              1200
        else
          dy=d(ns)
          ns=ns-1
        endif
        y=y+dy
13    continue
      return
```

```
      END
c
c                                                                    1210
      real*8 function CFfunction(mu)
      implicit none
      real*8 mu,Q,R,L,QRmu,QLmu,bessj1
      common/CFF/Q,R,L
c     print*, Q,R,L
      QRmu = Q*R*dsqrt(1.-mu**2.)
      QLmu = Q*mu*L/2.
      CFfunction = (dsin(QLmu)/QLmu)**2.*
     *          (2.*bessj1(QRmu)/QRmu)**2.
      return                                                         1220
      end
c
c
      real*8 function CF3function(mu)
      implicit none
      real*8 mu,Q,R,L,QRmu,QLmu,bessj1,QLtmu,F1,F2
      real*8 t, rhoo,rhoi,rhos
      common/CFF3/Q,R,L,t,rhoo,rhoi,rhos
      QRmu = Q*R*dsqrt(1.d0-mu**2.d0)
      QLtmu = Q*mu*(L+2.d0*t)/2.d0                                   1230
      QLmu = Q*mu*L/2.d0
      F1 = (rhoo-rhos)*dsin(QLtmu)/QLtmu
      F2 = (L/(L+2.d0*t))*(rhoo-rhoi)*dsin(QLmu)/QLmu
      CF3function = (F1+F2)**2.d0*
     *          (2.d0*bessj1(QRmu)/QRmu)**2.d0
      return
      end
c
c
      real*8 function func(prm, Q, j, xsectype, ii)                  1240
c
c This function is to convolute the calculated scattering intensity
c with given resolution function or not to
```

```
c
c
      implicit none
      character*20 xsectype
      integer i,j,resyn,nfold, ii
      real*8 prm(200), sum, reso, Q, QBE, xsec
      real*8 r1,r2,rl1,rl2,rlam,dlam,rdet, sig, QSTEP                    1250
      real*8 Qres
      common/resof1/reso(3,200,200),r1,r2,rl1,rl2,rlam,dlam,rdet,resyn
      common/resof2/sig(3,200),nfold
c
c
      if(resyn.eq.0) go to 1
      QBE=Q-3.*SIG(ii,J)
      IF(QBE .LE. 0.0) QBE=1.d-10
      QSTEP=6.*SIG(ii,J)/FLOAT(NFOLD)
      SUM=0.0                                                            1260
      DO 20 I=1,NFOLD
        Qres=QBE+(I-0.5)*QSTEP
        SUM=SUM+(XSEC(Qres,prm,xsectype)*RESO(ii,J,I))
 20   continue
      GOTO 2
 1    sum = xsec(Q, prm, xsectype)
 2    func = sum
      return
      end
c                                                                        1270
c
c
c
c
c This subroutine is trying to use anneal method to get the optimal fit
c of some functional data to experimental data
c
      subroutine anneal(edat,ftemp,pnumber,step,oldchi,state,xsectype)
c
```

```
c declare parameters used in the subroutine                              1280
c
      implicit none
      external func, ran3
      character*20 state,xsectype
      integer i, ii, accept, reject, estart(3), enumber, pnumber
      integer j, iloop, idum, imax, imequ, equ, iprint, sets
      integer etotal
      real*8 edat(3,4,200), prm, fdat(200),temp, step, oldchi, ran3
      real*8 chisqu, edge, func, chiave, outchi, IQ
      real*8 limit, sumchi, delchi, sumchi2, chidvi                      1290
      real*8 bound,thy,thyplum,dbound,p,ftemp
      real*8 ave, fave, outsum, avesum(200), favesum(200),chimin
      common/general/estart, enumber(3), prm(200), bound(2,200), imax
      common/QQQ/ave(200), fave(200), imequ, iprint, sets
ccccc
c
c assign initial values
c
      print *,state
      do 10 i=1, 200                                                     1300
        fdat(i) = 0.
        avesum(i) = 0.
        favesum(i) = 0.
        ave(i) = 0.d0
        fave(i) = 0.d0
 10   continue
c
c   First chisqu
      etotal = 0
      do 260 ii=1,sets                                                   1310
        etotal = etotal+enumber(ii)
 260  continue
      oldchi = 0.d0
      chimin = 1.d10
      do 250 ii=1,sets
```

```
      do 231 i=1, enumber(ii)
        fdat(i) = func(prm, edat(ii,1,i), i, xsectype, ii)
        oldchi = oldchi+
     +          ((fdat(i)-edat(ii,2,i))/edat(ii,3,i))**2.
231   continue                                                    1320
250  continue
     oldchi = oldchi/float(etotal)
     limit = 1.e-6
     iloop = 0
     edge = 0.
     accept = 0
     reject = 0
     idum = -100
     chiave = 1.e30
     outchi = 1.e30                                              1330
     sumchi = 0.
     sumchi2 = 0.
     equ = 0
     outsum = 0.
     if(state.eq.'continue') then
       open(unit=8,file='tmpprm.fit',status='unknown')
       rewind(unit=8)
       read(8,*) iloop
       read(8,*) oldchi
       read(8,*) outsum                                          1340
       read(8,*) sumchi
       read(8,*) sumchi2
       read(8,*) accept
       read(8,*) reject
       read(8,*) temp
       read(8,*) step
       read(8,*) equ
       read(8,*) iprint
       read(8,*) pnumber
       read(8,*) sets                                            1350
       read(8,*) prm(pnumber+1)
```

```
        do 12 i=1, pnumber
          read(8,*) prm(i), avesum(i),
     +              favesum(i)
 12     continue
        close(unit=8)
      endif
      open(unit=7, file='intave.fit',status='unknown')
      rewind(unit=7)
      open(unit=9, file='intprm.fit',status='unknown')                    1360
      rewind(unit=9)
      open(unit=11, file='intprmav.fit',status='unknown')
      rewind(unit=11)
      open(unit=13, file='intprmfa.fit',status='unknown')
      rewind(unit=13)
cccccc
c
c outer loop of all parameters
c
 1000 iloop = iloop+1                                                    1370
      temp = oldchi*ftemp
c
c inner loop of individual parameter
c
      do 100 j=1, pnumber
c
c update prm(j)
c
        p = prm(j)
        if(xsectype.eq.'complex'.and.j.eq.11) then                       1380
          bound(2,j) = bound(2,1)-prm(1)
        endif
          dbound = bound(2,j)-bound(1,j)
          if(dbound.eq.0.) go to 100
          thy = ran3(idum)
          thyplum = 2.*thy-1.
          prm(j) = prm(j)+thyplum*step*prm(j)
```

```
            if(prm(j).lt.bound(1,j)) then
              prm(j) = prm(j)+
     *         dbound*float(int((bound(1,j)-prm(j))/dbound)+1)
            endif
            if(prm(j).gt.bound(2,j)) then
              prm(j) = prm(j)-
     *         dbound*float(int((prm(j)-bound(2,j))/dbound)+1)
            endif
c
cccccc
c
c calculate chisquare of the current parameter set
c
      chisqu = 0.
      do 270 ii=1,sets
        do 20 i=1, enumber(ii)
          fdat(i) = func(prm, edat(ii,1,i), i, xsectype, ii)
          chisqu = chisqu+
     +             ((fdat(i)-edat(ii,2,i))/edat(ii,3,i))**2.
 20       continue
 270  continue
      chisqu = chisqu/float(etotal)
c
c accept the current parameter set or not by comparing the chisquare
c to the previous one, accept when smaller then the previous one, but
c also accept if exp(-chisq/temp) < random number(between 0 and 1)
c
      edge = ran3(idum)
      delchi = chisqu-oldchi
c
c
      if(delchi.le.0.) then
        accept = accept+1
        oldchi = chisqu
      else
        if(exp(-delchi/temp).gt.edge) then
```

```
          accept = accept+1
          oldchi = chisqu
        else
          reject = reject+1
          prm(j) = p
        endif
      endif                                                          1430
        avesum(j) = avesum(j)+prm(j)
        favesum(j) = favesum(j)+prm(j)
        outsum = outsum+oldchi
        sumchi = sumchi+oldchi
        sumchi2 = sumchi2+oldchi**2.
 100  continue
cccccc
c
c
        if(((float(iloop)/float(iprint))                             1440
     +       -int(float(iloop)/float(iprint))).eq.0.) then
          if(oldchi.le.chimin) then
            chimin = oldchi
            do 200 i=1,pnumber
              ave(i) = avesum(i)/float(iprint)
              fave(i) = favesum(i)/float(iloop)
 200          continue
              outchi = outsum/float(iloop*pnumber)
              chiave = sumchi/float(iprint*pnumber)
              chidvi = sumchi2/float(iprint*pnumber)-chiave**2.      1450
c
 2000           open(unit=8,file='tmpprm.fit',status='old')
              rewind(unit=8)
              write(8,*) iloop
              write(8,1023) oldchi
              write(8,1023) outsum
              write(8,1023) sumchi
              write(8,1023) sumchi2
              write(8,*) accept
```

```
            write(8,*) reject                                          1460
            write(8,1023) temp
            write(8,1023) step
            write(8,*) equ
            write(8,*) iprint
            write(8,*) pnumber
            write(8,*) sets
            write(8,1023) prm(pnumber+1)
            write(8,*) ' '
            do 21 i=1,pnumber
              write(8,1023) prm(i),                                    1470
     +                    avesum(i),favesum(i)
21           continue
1023          format(3e15.4)
            write(8,*) ' '
            do 280 ii=1,sets
              do 22 i=1,enumber(ii)
                IQ = func(prm, edat(ii,1,i), i, xsectype, ii)
                write(8,1023) edat(ii,1,i),IQ
22             continue
280           continue                                                 1480
            close(unit=8)
          endif
          sumchi = 0.d0
          sumchi2 = 0.d0
          do 30 i=1,pnumber
            avesum(i) = 0.d0
30         continue
          write(7,1024) iloop, chiave,chidvi,outchi
          write(9,1025) (prm(i),i=1,7)
          write(11,1025) (ave(i),i=1,pnumber)                          1490
          write(13,1025) (fave(i),i=1,pnumber)
1024       format(I11,3e11.3)
1025       format(7e11.3)
      endif
c
```

```
c check convergence or equilibration
c
      if(equ.ge.1) then
        if(iloop.ge.(10.*iprint).and.
     +       abs(outchi-chiave)/outchi.le.limit) then                    1500
          print *,'Out by Deviation'
          go to 9999
        endif
        if(oldchi.le.0.5) then
          print *,'Out by Chi Square'
          go to 9999
        endif
        if(iloop.eq.imax) then
          print *,'Out by Steps'
          go to 9999                                                      1510
        endif
      else
        if(iloop.eq.imequ) then
          iloop = 0
          equ = 1
          accept = 0
          reject = 0
          sumchi = 0.d0
          sumchi2 = 0.d0
          outsum = 0.d0                                                   1520
          do 25 i=1,pnumber
            avesum(i) = 0.d0
            favesum(i) = 0.d0
25        continue
          close(unit=7)
          close(unit=9)
          close(unit=11)
          close(unit=13)
          open(unit=7,file='intave.fit',status='unknown')
          rewind(unit=7)                                                  1530
          open(unit=9, file='intprm.fit',status='unknown')
```

```
            rewind(unit=9)
            open(unit=11, file='intprmav.fit',status='unknown')
            rewind(unit=11)
            open(unit=13, file='intprmfa.fit',status='unknown')
            rewind(unit=13)
          endif
        endif
cccccc
c                                                                    1540
      go to 1000
cccccc
c
 9999 close(unit=7)
      close(unit=9)
      close(unit=11)
      close(unit=13)
      return
      end
c                                                                    1550
c
      FUNCTION ran3(idum)
      INTEGER idum
      INTEGER MBIG,MSEED,MZ
      real*8 ran3
      REAL FAC
      PARAMETER (MBIG=1000000000,MSEED=161803398,MZ=0,FAC=1./MBIG)
      INTEGER i,iff,ii,inext,inextp,k
      INTEGER mj,mk,ma(55)
      SAVE iff,inext,inextp,ma                                       1560
      DATA iff /0/
      if(idum.lt.0.or.iff.eq.0)then
        iff=1
        mj=MSEED-iabs(idum)
        mj=mod(mj,MBIG)
        ma(55)=mj
        mk=1
```

```
      do 11 i=1,54
        ii=mod(21*i,55)
        ma(ii)=mk                                                    1570
        mk=mj-mk
        if(mk.lt.MZ)mk=mk+MBIG
        mj=ma(ii)
11     continue
      do 13 k=1,4
        do 12 i=1,55
          ma(i)=ma(i)-ma(1+mod(i+30,55))
          if(ma(i).lt.MZ)ma(i)=ma(i)+MBIG
12       continue
13     continue                                                      1580
      inext=0
      inextp=31
      idum=1
    endif
    inext=inext+1
    if(inext.eq.56)inext=1
    inextp=inextp+1
    if(inextp.eq.56)inextp=1
    mj=ma(inext)-ma(inextp)
    if(mj.lt.MZ)mj=mj+MBIG                                           1590
    ma(inext)=mj
    ran3=mj*FAC
    return
    END
c
c
    real*8 FUNCTION SIGQ(X)
    implicit none
    integer resyn, nfold
    real*8 pi,r1,r2,rl1,rl2,rlam,dlam,rdet,wk0,sigdet,x,thet0         1600
    real*8 aa1,a2,bet1,delq1,reso,sig
    common/resof1/reso(3,200,200),r1,r2,rl1,rl2,rlam,dlam,rdet,resyn
    common/resof2/sig(3,200),nfold
```

```
c
      PI=3.1415927
c
c        r1=1.9
c        r2=0.545
c        rl1=1636.
c        rl2=127.73                                                    1610
c        rlam=6.
c        dlam=0.147
c        rdet=0.39
c
      WK0=2.*PI/RLAM
      SIGDET=RDET*WK0/RL2*0.4246609
      THET0=dASIN(X/2./WK0)
      AA1=R1/(RL1+RL2/dCOS(2.*THET0)**2)
      A2=R2*dCOS(2.*THET0)**2./RL2
C                                                                       1620
      IF(A2.le.AA1) then
C
          BET1=2.*R1/RL1-0.5*R2*R2*(dCOS(2*THET0))**4.*(RL1
     !       +RL2/dCOS(2.*THET0)**2)**2./RL1/RL2/RL2/R1
      else
C
        BET1=2.*R2*(1./RL1+COS(2.*THET0)**2/RL2)
     !       -0.5*R1*R1*RL2/(R2*dCOS(2.*THET0)**2.*
     !       RL1*(RL1+RL2/dCOS(2.*THET0)**2))
C                                                                       1630
      endif
C
      DELQ1=dSQRT((WK0*dCOS(THET0)*BET1)**2.+(X*DLAM)**2.)
C
      SIGQ=(0.4246609*DELQ1)**2.+SIGDET**2.
      RETURN
      END
c
c
```

```
      real*8 FUNCTION FRES(SIGQ,QCEN,Q)                                  1640
      implicit none
      real*8 x,sigq,qcen,q,t,bess,fresx,bessx
      X=QCEN*Q/SIGQ
      T=X/3.75
      IF(ABS(X).le.3.75) then
        BESS=1.+3.5156229*T*T+3.0899424*T**4.+1.2067492*T**6.
     !       +0.2659732*T**8.+0.0360768*T**10.+0.0045813*T**12.
        FRESX=Q/SIGQ*EXP(-0.5*(QCEN*QCEN+Q*Q)/SIGQ)*BESS
      else
        BESSX=(0.39894228+0.01328592/T+0.00225319/T**2.                  1650
     !         -0.00157565/T**3.+0.00916281/T**4.-0.02057706/T**5.
     !         +0.02635537/T**6.-0.01647633/T**7.+0.00392377/T**8.)
        FRESX=Q/SIGQ*EXP(-0.5*(QCEN*QCEN+Q*Q)/SIGQ+X)*BESSX/dSQRT(X)
      endif
      FRES=FRESX
      RETURN
      END
c
c                                                                        1660
```

Bibliography

[1] A. Guinier, *Nature(London)* **142** (1938) 569-570.

[2] G. Porod, *Acta Phys. Austriaca* **2** (1948) 255-292.

[3] O. Kratky, *Z. Elektrochem.* **58** (1954) 49-53.

[4] V. Luzzati, *Acta Crystallogr.* **13** (1960) 939-945.

[5] P. W. Schmidt, *Acta Crystallogr.* **19** (1965) 938-942.

[6] A. Guinier and G. Fournet, *Small-Angle Scattering of X-Rays* (1955) Wiley, New York.

[7] O. Glatter and O. Kratky, eds. *Small-Angle X-Rays Scattering* (1982) Academic Press, London.

[8] L. A. Feigin and D. I. Svergun, *Structure Analysis by Small-Angle X-Ray and Neutron Scattering* (1987) Plenum Press, New York.

[9] P. W. Schmidt and B. A. Fedorov, *J. Appl. Cryst.* **11** (1978) 411-416.

[10] O. Glatter, *J. Appl. Cryst.* **10** (1977) 415-421.

[11] P. W. Schmidt, *J. Appl. Cryst.* **21** (1988) 602-612.

[12] V. Ramakrishnan, *J. Appl. Cryst.* **18** (1985) 42-46.

[13] G. D. Wignall, D. K. Christen, and V. Ramakrishnan, *J. Appl. Cryst.* **21** (1988) 438-451.

[14] W. Schmatz, T. Springer, J. Schelten, and K. Ibel, *J. Appl. Cryst.* **7** (1974) 96-116.

[15] P. A. Seeger, *Nucl. Instrum. Methods* **178** (1980) 157-161.

[16] P. A. Seeger, *Physica (Utrecht)* **136B** (1986) 106-109.

[17] D. F. R. Mildner and J. M. Carpenter, *J. Appl. Cryst.* **17** (1984) 249-256.

[18] D. F. R. Mildner and J. M. Carpenter, *J. Appl. Cryst.* **20** (1987) 419-424.

[19] P. A. Seeger and R. Pynn, *Nucl. Instrum. Methods* **A245** (1986) 115-124.

[20] J. A. Miller, S. L. Cooper, C. C. Han, and G. Pruckmayr, *Macromolecules* **17** (1984) 1063-1071.

[21] D. F. R. Mildner, J. M. Carpenter and D. L. Worcester, *J. Appl. Cryst.* **19** (1986) 311-319.

[22] R. P. Jr Hjelm, *J. Appl. Cryst.* **21** (1988) 618-628.

[23] P. B. Moore, *J. Appl. Cryst.* **13** (1980) 168-175.

[24] R. W. Hendricks and P. W. Schmidt, *Acta Phys. Austriaca* **26** (1967) 96-122.

[25] R. W. Hendricks and P. W. Schmidt, *Acta Phys. Austriaca* **37** (1973) 20-30.

[26] T. Freltoft, J. K. Kjems, and S. K. Sinha, *Phys. Rev. B* **33** (1987) 269-275.

[27] J. S. Pedersen, D. Posselt, and K. Mortensen, *J. Appl. Cryst.* **23** (1990) 321-333.

[28] N. Holt, *Rev. Sci. Instrum.* **28** (1957) 1-3.

[29] K. Ibel, *J. Appl. Cryst.* **9** (1976) 296-309.

[30] S. H. Chen, S. L. Chang, and R. Strey, *J. Chem. Phys.* **93** (1990) 1907.

[31] M. Kotlarchyk, S. H. Chen, J. S. Huang, and M. W. Kim, *Phys. Rev.* **A29** (1984) 2054.

[32] M. Kotlarchyk, S. H. Chen, J. S. Huang, and M. W. Kim, *Phys. Rev. Lett.* **53** (1984) 914.

[33] L. Auvray, J. P. Cotton, R. Ober, and C. Taupin, *J. Phys. Chem.* **88** (1984) 4586.

[34] F. Lichterfeld, T. Schumeling, and R. Strey, *J. Phys. Chem.* **90** (1986) 5762.

[35] S. H. Chen, S. L. Chang, R. Strey, J. Samseth, and K. Mortensen, *J. Phys. Chem.* **95** (1991) 7427.

[36] J. Rouch, A. Safouane, P. Tartaglia, and S. H. Chen, *J. Chem. Phys.* **90** (1990) 3756.

[37] C. Cametti, P. Codastefano, P. Tartaglia J. Rouch, and S. H. Chen, *Phys. Rev. Lett.* **64** (1990) 1461.

[38] G. S. Grest, I. Webman, S. Safran, and A. L. R. Bug, *Phys. Rev.* **A33** (1986) 2842.

[39] M. Lagues, *J. Phys.(Paris)Lett.* **40** (1979) L331.

[40] J. P. Clerc, G. Giraud, J. M. Laugier, and J. M. Luck, *Advances in Phys.* **39** (1990) 191.

[41] R. J. Baxter, *J Chem. Phys.* **49** (1968) 2770.

[42] J. K. Percus and G. J. Yevick, *Phys. Rev.* **110** (1958) 1.

[43] B. Barboy, *J. Chem. Phys.* **61** (1974) 3194.

[44] A. Coniglio, U. De Angelis, and A. Forlani, *J. Phys.* **A10** (1977) 1123.

[45] Y. C. Chiew and E. D. Glandt, *J. Phys.* **A16** (1983) 2599.

[46] E. Y. Sheu and S. H. Chen, J. S, Huang, and Y. C. Sung, *Phys. Rev.* **A39** (1989) 5867.

[47] C. Robertus, W. H. Philipse, J. G. H. Joosten, and Y. K. Levine, *J. Chem. Phys.* **90** (1989) 4482.

[48] M. Kotlarchyk and S. H. Chen, *J. Chem. Phys.* **79** (1983) 2461.

[49] S. V. G. Menon, V. K. Kelker, and C. Manohar, *Phys. Rev.* **A43** (1990) 1130.

[50] P. Tartaglia, J. Rouch, and S. H. Chen, *Phys. Rev.* **A45** (1992) 7257.

[51] S. H. Chen and J. Teixeira, *Phys. Rev. Lett.* **57** (1986) 2583.

[52] D. Stauffer, *Phys. Report* **54** (1979) 1.

[53] J. Rouch, P. Tartaglia, and S. H. Chen, *Phys. Rev. Lett.* **71** (1993) 1947.

[54] S. H. Chen, F. Mallamace, J. Rouch, and P. Tartaglia, in *"Slow Dynamics in Condensed Matter"* K. Kawasaki, T. Kawakatsu and M. Tokuyama eds., American Institute of Physics, New York, p. 301, 1992.

[55] S. H. Chen, J. Rouch, and P. Tartaglia, *Croatica Chemica Acta* **65** (1992) 353.

[56] J. B. Gilmour, J. O. Zwicker, J. Katz, and R. L. Scott, *J. Phys. Chem.* **71** (1967) 3259.

[57] R. L. Scott, *J. Phys. Chem.* **62** (1958) 136.

[58] D. G. Le Grand and G. L. Gaines, *J. Colloid Interface Sci.* **50** (1975) 272.

[59] P. Mukerjee, *J. Am. Oil Chem. Soc.* **59** (1982) 573.

[60] P. LoNostro, C. Y. Ku, S. H. Chen, and J. S. Lin, *J. Phys. Chem.* **99** (1995) 10858–10864.

[61] P. LoNostro, *Advances Colloid Interface Sci.* **56** (1995) 245-287.

[62] J. Höpken and M. Möller, *M. Macromolecules* **25** (1992) 2482.

[63] C. E. Hoyle, D. Kang, C. Jariwala, and A. C. Griffin, *Polymer* **34** (1993) 3070.

[64] J. Höpken, C. Pugh, W. Richtering, and M. Möller, *Makromol. Chem.* **189** (1988) 911.

[65] D. L. Dorset, *Macromolecules* **23** (1990) 894.

[66] V. Percec, D. Tomazos, A. E. Feiring, *Polymer* **32** (1991) 1897.

[67] J. F. Rabolt, T. P. Russell, R. J. Twieg, *Macromolecules* **17** (1984) 2786.

[68] T. P. Russell, J. F. Rabolt, R. J. Twieg, R. L. Siemens, B. L. Farmer, *Macromolecules* **19** (1986) 1135.

[69] C. Viney, T. P. Russell, L. E. Depero, R. J. Twieg, *Mol. Cryst. Liq. Cryst.* **168** (1989) 63.

[70] J. Höpken, *Fluorocarbon-Hydrocarbon Mixtures.* Ph. D. Thesis, University of Twente, The Netherlands, (1991).

[71] M. P. Turberg, J. E. Brady, *J. Am. Chem. Soc.* **110** (1988) 7797.

[72] P. Lo Nostro, S. H. Chen, *J. Phys. Chem.* **97** (1993) 6535.

[73] C. Viney, R. J. Twieg, and T. P. Russell, *Mol. Cryst. Liq. Cryst.* **182B** (1990) 291.

[74] G. L. Gaines, *Langmuir* **7** (1991) 3054.

[75] M. S. B. Munson, *J. Phys. Chem.* **68** (1964) 796.

[76] J. Hildebrand, B. B. Fisher, and H. A. Benesi, *J. Am. Chem. Soc.* **72** (1950) 4348.

[77] V. P. Skripov and M. Z. Faizullin, *J. Chem. Thermodyn.* **21** (1989) 687.

[78] C. L. Young, *Trans. Faraday Soc.* **65** (1969) 2639.

[79] A. J. B. Cruickshank, B. W. Gainey, and C. L. Young, *Trans. Faraday Soc.* **64** (1968) 337.

[80] C. Y. Ku, S. H. Chen, J. Rouch, and P. Tartaglia, *Int. J. Thermophys.* **16** (1995) 1119-1134.