ABSTRACT

LAMINAR VISCOUS RESUSPENSION IN TWO-DIMENSIONAL CHANNELS

by
Wenyue Yin

The laminar viscous resuspension of sphere particles in two-dimensional channel with gravity and electrostatic effects is studied numerically. The constitutive equation for particles include the shear-induced diffusion flux, gravitational sedimentation flux and electro-static diffusion flux. In addition the continuity equation and the momentum equations for incompressible flow are solved for developing flow using finite element method.

The parameters in this study are ratio of particle diameter to channel height d/h, Reynolds number Re, Froude number Fr, uniform inlet particle volume fraction ϕ_o, and electrostatic parameter Q. The numerical simulation included d/h from 0.03 to 0.12, Re from 5 to 46, Fr from 0.048 to 0.38, ϕ_o from 0.20 to 0.35 and Q from 0 to 0.005.

It was found that there may exist three maximum particle concentrations; between the channel centerline and the top channel wall, along the centerline and between the centerline and the bottom channel wall. However due to gravity effect the highest particle concentration is at the bottom wall. The maximum concentration near the top wall increases with increasing ϕ_o for $0.3>\phi_o>0.20$. At $d/H=0.03$ and $\phi_o=0.30$ this maximum is $\phi/\phi_o = 1.40$. This maximum concentration also increases with decreasing d/H ratio.

The electrostatic force tends to diminish the maximum concentration near the top wall but enhance the maximum near the centerline. It was found that the electrostatic

effect may be neglected when $Q<0.0001$ and may become significant when $Q>0.001$.

The velocity profile of the fluid deviates from the Poiseuille flow (parabolic) profile in the top portion of the channel. At the midway between the centerline and the top wall the velocity is about 11 % higher than that of parabolic profile.

Comparison with some experimental results for flow in a circular tube gives qualitatively good agreement.

LAMINAR VISCOUS RESUSPENSION IN TWO-DIMENSIONAL CHANNELS

by
Wenyue Yin

A dissertation
Submitted to The Faculty of
New Jersey Institute of Technology
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Department of Mechanical Engineering

May 2000

APPROVAL PAGE

LAMINAR VISCOUS RESUSPENSION IN TWO-DIMENSIONAL CHANNELS

Wenyue Yin

Dr. Rong Chen, Dissertation Advisor — Date
Professor of Mechanical Engineering, NJIT

Dr. Jonathan H. C. Luke, Committee Member — Date
Associate Professor of Mathematical Sciences, NJIT

Dr. Ernest S. Geskin, Committee Member — Date
Professor of Mechanical Engineering, NJIT

Dr. Kwabena A. Narh, Committee Member — Date
Assistant Professor of Mechanical Engineering, NJIT

Dr. Pushpendra Singh, Committee Member — Date
Assistant Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Wenyue Yin

Degree: Doctor of Philosophy

Date: May 2000

Undergraduate and Graduate Education:

- Doctor of Philosophy in Mechanical Engineering
  New Jersey Institute of Technology, Newark, NJ, USA, 2000

- Master of Science in Mechanical Engineering
  Tsinghua University, Beijing, P. R. China, 1996

- Bachelor of Science in Thermal Engineering
  Tsinghua University, Beijing, P. R. China, 1993

Major: Mechanical Engineering

Presentations and Publications:

L. Sridhar, Wenyue Yin, and K. A. Narh,
"The effect of shrinkage induced interface gap on the thermal contact resistance between the mold and plastic in injection molding."
Journal of Injection Molding Technology, March 2000, Vol. 4, NO. 1, 44-49.

L. Sridhar, Wenyue Yin, and K. A. Narh,
"The effect of contacting material thermal conductivity on the computation of gap and contact thermal resistance."
ASME National Heat Transfer Conference, Albuquerque, NM, 1999.

This dissertation is dedicated to
my family for their love and support

ACKNOWLEDGMENT

The author would like to express her sincere thanks and deepest gratitude to her advisor, Dr. R. Chen, for the remarkable guidance, constant support, encouragement and patience throughout her study.

The author would also like to express her thanks and appreciation to Drs. Luke, Geskin, Singh and Narh for their valuable suggestion to the dissertation and for serving as the committee members.

The author appreciates the financial support from the Department of Mechanical Engineering, NJIT, in the form of assistantships during the author's doctoral program.

The author also would like to acknowledge Dr. Z. Li, Mr. J. Guo and Mr. W. Lee for their help and suggestions to this study.

Finally, the author would like to express her deepest gratitude to her family members, especially her husband Simin Yao. Without his firm support and encouragement, the work could not be done smoothly.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

1.1 General Introduction and Practical Application

Suspensions are very common phenomenon in our daily life. Most biological flows belong to this category. Blood or milk consist of a liquid and suspended solid particles of various kinds. Similarly, most paints are suspensions of a solid pigment in a suitable liquid base [1]. Suspensions are also utilized in a very wide variety of applications, such as hydraulic fracturing technology, ceramics, reinforced polymer composites, and transport of slurries. To successfully process the suspensions in engineering applications, the most important concern is to be able to predict the particle concentration and motion at any location. For this reason, a great deal of attention from both practical and the fundamental points of view has been generated in recent years..

Among the many kinds of suspensions, one suspension system appears to have been studied in most detail: the macroscopically homogeneous materials, especially very small particles freely suspended in Newtonian fluid. For this kind of suspension system, we can assume it as effective continua on a macroscale.

After decades of study on this system by many investigations [9][10][16], it has been recognized that the flowing suspension exhibits gross particle movement, which is called "migrate" within the flow. This migrate can be observed in the concentrated suspension undergoing shear, a flux of particles will be induced from regions of high shear to low and from regions of high particle concentration to low. The reason for this migrate was studied by several authors, they identified three causes of particle migration migrate was studied by several authors, they identified three causes of particle migration and diffusion: shear-rate gradients, relative viscosity gradients and concentration gradients. In these studies they consider only flows of large Peclet number, so the classical diffusion is negligible. We call this particle migration due to shear-rate gradients and particle motion due to viscosity or concentration gradients as shear-induced migration.

This shear induced migration has important implications on many practical application. Here are some examples:

Measurement of viscosity [2] :Before the discovery of the shear-induced migration, the viscometric measurement on suspensions using different devices were found to obtain values for the effective viscosity as a function of the volume concentration ϕ which differs from one another by as much as an order of magnitude at the high concentration. Now, when we measure the effective viscosity of concentrated suspensions by the viscometer, we must consider the shear induced non-uniformity in particle concentration within the device, otherwise, it will be subjected to serious errors.

Manufacturing of solid-rocket propellants [3] : For instance, the performance of rocket motors can be enhanced by reproducibly casting propellant mixes to yield a uniform composition through the motor, so that the motor can provide uniform burn rate and mechanical properties. But in reality, the burn rate and mechanical properties in motor are moderately non-uniform, and depend on the location within the motor as well as the orientation of the sample. Some analysis suggest that flow-induced particle alignment and segregation are responsible for the variability in properties across the motor. To solve this problem, we need a better understanding of suspension rheology and flow induced microstructure changes in suspensions.

Hydraulic fracture [4] : When a hydraulic fracture is created, a fluid containing proppant particles is pumped into it. Ideally, the proppant particles should settle evenly along the entire length of the fracture so that when the pump ceases, the fracture is wedged open by the sedimented particles. By using the viscous resuspension effect, it is possible to entrain particles from a settled bed back into the bulk shear flow which will enable them to be convected deep into the fracture channel and thus avoid the possibility of closure.

1.2 Fundamental of Suspension

a. Viscous drag of a single particle

If the particle velocity is different from the velocity of fluid, the force exerted by the gas on the particle depends on the relative velocity. This force is called drag force.

At very low relative velocity, Stokes suggested that the inertial effects were so small that can be neglected from the Navier-stokes equation. His asymptotic approximation thus obtained gives a symmetric flow field about a sphere. The resulting drag force is[5]

$$F=6 \pi \mu a U \tag{1.1}$$

where μ is the dynamic viscosity of the fluid material. Two-third of F is the shear force (friction drag), and one third of F is due to the pressure force (form force).

With the definition of the drag coefficient, C_D,

$$C_D = F / (\pi a^2)(\rho U^2 / 2) \tag{1.2}$$

We get, Stokes' law

$$C_D = 24/\mathrm{Re} \quad (1.3)$$

where the Reynolds number, Re, is defined as

$$\mathrm{Re} = 2\, a\, \rho U/\, \mu \quad (1.4)$$

where ρ is the density of the fluid material, and μ is the dynamic viscosity of the fluid material.

b. Gravitational settling

When a particle settles in fluid under the influence of gravity, the particle starts its motion as free fall, and as its velocity increases, the drag term increases and opposes the motion. During this settling process, the most important quantity is the terminal velocity, V_s

$$V_s = \frac{2}{9}\frac{(\rho_p - \rho)ga^2}{\mu} \quad (1.5)$$

where ρ_p is the density of particle.

c. Effective viscosity of suspension

For a long time, the theoretical study of suspension has been limited to dilute concentration (typically the volume fraction is less than 3-4%) such that the suspensions behave like Newtonian fluid. The effective viscosity of this suspension, μ^*, is given by Einstein for the case of solid spheres[6]

$$\frac{\mu^*}{\mu} = 1 + \frac{5}{2}\phi + O(\varphi^2) \quad (1.6)$$

This expression is exact under the very dilute suspension, and has formed the basis for much of the theoretical work on the subject since the turn of the century.

1.3 Theme and Summary of This Study

The motive of this study is to obtain the knowledge of viscous suspension behavior in the two dimensional channel by finite element analysis.

In this study, a well mixed suspension with uniform velocity will be induced into a two dimensional channel. Under the effect of shear-induced migration, and gravitational migrate, the concentration and the velocity of the suspension will be changed. This will be a transitional process, and at the entrance of the channel, the suspension flow is a developing flow.

The suspension system is treated as an effective continuum Newtonian flow, and a series of governing equations will be built, which include the Navier-stokes equation for two dimensional flow, and the concentration distribution equation for suspension. There are four parameters in the governing equations, they are Reynolds number, the modified Froude number, the relative density ratio of the suspended particles to the suspending fluid, and the square of the ratio of the particle size to the characteristic length scale.

Due to the complexity of the governing equation, the problem can only be solved by numerical method. In our study, we will use the finite element method to solve the equations. Matlab and C++ programs are written to carry out the calculations. The influence of different parameters will be studied, and the results of this study will be discussed in detail.

In chapter two, a literature survey of many previous investigation is introduced in order to obtain a general understanding on flow of suspensions. In chapter three, the principle and algorithm of constructing a viscous suspension and the governing equations for flow of suspensions in two-dimensional channel are presented. In chapter four, the numerical analysis and calculation of the particle concentration and flow velocity are carried out with the finite element method. In chapter five, the results of particle concentration and floe velocity under different conditions are discussed and various dimensionless parameters are analyzed accordingly. The conclusions and suggestions for further investigation are presented in chapter six.

CHAPTER 2

LITERATURE SURVEY

As mentioned in the previous chapter, many researchers have conducted investigations on viscous suspension flow. Due to the complexity of the problem, a large number of publications in both analytical and experimental aspects of many facets have been released. Obviously, it is impossible to review all these publications in this chapter. A brief survey that covers studies of shear induced migration and related investigation of viscous suspension flow is presented here.

Einstein gave the formula for effective viscosity in very dilute suspension in 1906[6], and this formula was considered as the foundation of many studies later on. It led to the very commonly accepted view that suspensions should behave as Newtonian fluids for all values of the particle concentration up to the maximum.

In 1972, Batchelor and Green [7] derived the rigorous extension of Einstein's formula to $O(\phi^2)$ for suspensions containing monodisperse solid spheres. Their paper is particularly significant on various counts: (1) It described a systematic procedure, subsequently extended to more completed systems. (2) It further illustrated the application of a renormalization technique, for recasting certain seemingly divergent expressions, for such bulk quantities as the particle stress, into expressions involving convergent integrals; and (3)It clearly demonstrated that such suspensions do not behave as effective Newtonian fluids as has been generally supposed, in that the numerical value of the coefficient of the $O(\phi^2)$ term in Eq. (1.6) was shown to depend on the type of flow to which the suspension is being subjected. They uncovered this non-Newtonian feature by their rigorous calculation, but this was realized only recently.

During the course of their study of the rheology of concentrated suspensions of neutrally buoyant spheres in Newtonian fluids, Gadala-Maria and Acrivos[8] reported their puzzling observation. At very high particle concentrations (ϕ.0.40), the effective viscosity of the suspension, which is measured in the conventional Couette viscometer, decreased slowly with prolonged shearing and finally reached an equilibrium value, and this value would remain time independent. The relative magnitude of this decrease was quite large, that is about a factor of 2 for 45% suspension, and it was a strong function of concentration. The suspension fluid is Newtonian and there is no degradation under experimental condition, the particle size is large enough that the nonhydrodynamic effects, e.g., Brownian motion effects, surfaces forces, and electroviscous forces would not play a significant role, and the total strain or fluid deformation associated with the viscosity decrease is very large. Under these conditions, the suspension should behave as incompressible, isotropic, Newtonian fluids, i.e., fluids for which the bulk stress is a linear function of the bulk rate of strain with the constant of proportionality, the so-called effective viscosity, being a scalar. But, the results of their experiments, which appeared to be novel, strongly suggested that even suspension with relatively large spheres could develop an anisotropic structure when sheared and consequently, a single scalar parameter -- the effective viscosity -- might no longer suffice to relate the stress to the rate of strain.

David Leighton and Andreas Acrivos[9] repeated the experiment done by Gadala-Maria and Acrivos, and they found that the observed decrease in viscosity results from a migration of particles out of the sample being sheared in the gap between the bob and the cup of the Couette device and into the reservoir containing the stagnant part of the suspension. Clearly, such migrations would tend to decrease the concentration in the gap, and since in concentrated suspensions the viscosity is a very strong function of ϕ. Its measured value can be significantly affected by even small variation in the particle concentration. They also discovered a new phenomenon -- a short term increase in the viscosity upon initial shearing of a suspension in a Couette device. This suggested that in a concentrated suspension flowing in a channel, particles would migrate owing to gradients in shear stress. Therefor, they provided the first evidence for the existence of a non-uniform equilibrium concentration distribution in a suspension undergoing Poiseuille flow at zero Reynolds number. After eliminate the possibility of gravity and inertial effects, they explained the mechanism responsible for this particle migration as being due to a shear-induced particle drift arising from irreversible interparticle interactions. They pointed out that these interparticle interactions lead to effectively diffusivities both normal to the plane of shear and normal to the direction of fluid motion within the plane of shear, and these interactions should induce, within a shear flow, particle drift from regions of high to low shear stress. This process of particle drift were estimated to be of sufficient intensity to account for the observed initial viscosity increase.

Leighton and Acivos[10] continued their research on the mechanism of shear-induced resuspension. They pointed out that a single marked particle immersed in a viscous suspension of otherwise identical particles undergoing the simple shear flow, would suffer a series of displacements normal to the fluid streamline if it interacted with the other particles in suspension. Taken together, these displacements constituted a random walk. Moreover, if the concentration distribution of particles in the suspension is at equilibrium, the particle displacement due to this random walk will have zero means.

They called this process shear-induced self-diffusion and developed a relationship between the shear-induced coefficient of self-diffusion parallel to gradients in fluid velocity and variations in the time it takes for a marked particle immersed in the suspension to complete a circuit of a Couette device. After that they designed an experiment to measure the successive times taken by a marked particle to complete a circuit of the Couette device to determine the shear-induced coefficient of self-diffusion of spheres at low Reynolds number. They found that the diffusion coefficient was proportional to the shear rate and the square of the particle radius and was an increasing function of concentrate ϕ, approximately equal to $0.5\phi^2$ at low concentration.

Leighton and Acrivos[11] demonstrated the viscous resuspension in their later research. The viscous resuspension is the process, in the presence of a shear flow, an initially settled layer of negatively buoyant particles is entrained into the bulk fluid and is convected away under very low Reynolds number. This resuspension was characterized by the equilibrium height achieved by the initially settled bed of particles expanding in the presence of an applied shear stress. They explored the possibility that the resuspension was actually due to the shear-induced diffusion of the particles balancing their downward flux due to settling. They further developed a relationship between the increase in height of the settled layer and an integral involving the shear-induced diffusion coefficient they defined before by balancing the particle flux.

$$\frac{h - h_0}{a} = \frac{9}{2}\frac{\tau}{ag\Delta p}\int_0^{\phi_0}\frac{\phi_0 - \phi}{\phi_0}\frac{\hat{D}}{\phi f \mu_\tau}d\phi \tag{2.1}$$

The change in height of the settled layer when sheared, made dimensionless with the particle radius α, should be proportional to an integral involving the shear-induced diffusion coefficient $\hat{D}$, the settling hindrance function f and the shear viscosity μ_τ.

with approximation of $\hat{D} = \frac{1}{3}\phi^2(1 + \frac{1}{2}e^{8.8\phi})$

and $f\mu_\tau = (1-\phi)$

and assuming a maximum concentration $\phi_0 = 0.58$ for the settled layer, they got an equilibrium height of 2. Compared to the experimental results of 2.1, their hypothesis that viscous resuspension results from a balance between shear-induced diffusion and settling due to gravity is consistent with the experimental evidence.

After Acrivos and Leighton published their papers in 1986 and 1987, many researchers began to do study on this field, they worked on both experimental and theoretical research, and made a lot of progress, these made us have a better understand of the phenomenon.

Acrivos, Mauri and Fan [15] then conducted an experiment to exam the theory prediction of the height of resuspension developed by Acivos and Leighton.

Based on Acrivos and Leighton's analysis, they obtained the ratio of the resuspension height h to the initial height of settled layer h_0

$$\frac{h - h_0}{h_0} = A \int_0^{\phi(0)} \frac{\hat{D}(\phi)}{f(\phi)} \left(\frac{1}{\phi} - \frac{1}{\phi_0}\right) d\phi \qquad (2.2)$$

with $A = \frac{2}{9}\frac{\mu_0 \dot{\gamma}}{g \Delta p h_0}$

this clearly showed that, for a given choice of $f(\phi)$, $\hat{D}(\phi)$, $\mu_r(\phi)$ and ϕ_0, the relative change in the resuspension height is a function only of A.

They measured the height of the resuspended layer of an initially settled bed of particles undergoing a laminar shear flow in a narrow-gap Couette device, and compared with their theoretical predictions. They found that the experimental data for $\Delta h / h_0$ as a function of A showed excellent quantitative agreement with the prediction. They found that the wall slip effect was very small and could be negligible.

In 1991, A. L. Graham et. al [12] reported their experimental study on shear-induced diffusion. They described nuclear magnetic resonance (NMR) imaging of two concentrated suspensions undergoing flow between rotating concentric cylinders (wide-gap, annular Couette flow). They studied on both monodisperse (50% by volume) and bidisperse (60% volume) suspensions. The NMR imaging of the two suspensions shows that substantial particle migration away from the inner rod began to happen immediately upon rotation of the inner rod. After ten turns, the major intensity change took place close to the inner boundary of the annular space. There was an increase of the fluid fraction near the center rod and a corresponding decrease near the center of the gap. After 50 rotations, movement of the fluid even from the outermost region of the sample was seen. It means that the particles migrate from higher shear rate regions near the rotating inner cylinder to the lower shear rate regions near the stationary outer wall, thus established large concentration gradient. In addition, the large particles in the bimodal suspension form concentric cylindrical sheets, parallel to the axis of the Couette device. In their studies, there was no significant axial migration of the particles was observed: the dispersion of particles was almost entirely in the radial direction.

Sinton and Chow [3] demonstrated again that NMR flow imaging can be used to study the solid-fluid suspension. They found that for a Poiseuille flow of suspension with high volume fraction, the velocity profile was different from the velocity parabolic distribution. The velocity distribution appeared to be blunted with slower than expected velocities toward the center of the pipe for the 42% and 50% volume fraction suspension, while the velocity distribution of suspension of 21% volume fraction exhibited excellent agreement with the parabolic shape. This blunted velocity profile was consistent with prediction calculated by Leighton. Also, in their experiments on suspensions, the normalized entrance length (length needed for particle concentration across the pipe to reach steady state) was much larger the normalized transition length (length needed to established a fully developed flow) for Newtonian fluid.

NMR has become an important technique to study the behavior of suspension. Altobelli, Givler and Fukushima [17] measured the non-uniform velocity and concentration distribution for suspension of small, negatively buoyant particles flowing in horizontal tubes. All the measurements were performed during statistically steady flow condition with average fluid velocities up to 25cm/s. The mean particle concentration $\bar{\phi}_s$ ranged between 0.0 to 0.39. They recorded the time-average measurements by using a three-dimensional, flow-compensation, NMR technique. They observed that for slow flows, the suspension exhibited stratified flow, a region of clear fluid flows above a highly concentrated bed, and the thickness of the clear fluid layer decreased as $\bar{\phi}_s$ increased. Comparison among images for flowing versus static experiments showed that even low flow rates caused the particle bed to dilate and consequently become resuspended. For $\bar{\phi}_s$=0.39, they found that the particle volume fraction was approximately equal to the maximum packing fraction in the static samples at some regions that were slightly above the tube centerline. Velocity profiles of the suspension with $\bar{\phi}_s$=0.39 were blunt and shifted slightly upward. The particle-rich regions correlated with the flattest region of the velocity profile. As concentration or velocity increased, both the fluid velocity and volume fraction profiles became more axisymmetric.

Abbott et al. [18] used the NMR imaging to observe the evolution of radial concentration and velocity profiles of initially well-mixed concentrated suspensions of spheres in viscous Newtonian liquids undergoing flow between rotating concentric cylinders. They found that in Couette flow, particles migrate from high shear-rate region near the inner rotating cylinder to the low shear-rate region at the outer wall. The particle concentration near the outer wall approached the maximum packing for randomly distributed spheres at steady state, and velocity profiles revealed the suspension is almost stagnant in these regions. They also examined the functional dependence of particle migration in concentrated suspension undergoing shear flow in a wide-gap Couette. The primary experimental parameters were strain, shear rate, viscosity of the suspending liquid, particle diameters, and degree of polydispersivity of the particle phase. They showed that the migration rate was directly proportional to the shear and independent of the suspension liquid viscosity. The particle migration rate did depend on the particle diameter raised to the ~2.6 power for suspensions where ϕ =50%. These observations implied that the particle migration was not a function of the sphere's Reynolds number. The migration rate was also found to depend weakly on the polydispersivity of the particulate phase.

B. K. Chapman and D. T. Leighton Jr. [13] utilized dynamic measurements of the response of a suspension of negatively buoyant non-colloidal spheres to a step change in the applied shear rate and used the time-dependent variation in observed viscosity to estimate the shear-induced effective diffusivity. By employing the model that resuspension is resulted from the balance between the downward flux of particles due to sedimentation and an upward flux due to a shear-induced effective diffusion, they posed the governing equation for time-dependent concentration distribution. After making some simplification and assumption, they obtained the analytic solution to the equation, thus got the expression for time-dependent change in the dimensionless observed viscosity. The transit behavior only depended on the diffusivity and several measured parameters, so the diffusivity can be determined from transit viscosity measurements using the expression easily. From the determination of diffusivity from experiments and calculation, it is also easy to calculate the hindered settling coefficient, in the plane of shear. They expanded the diffusivity in a Taylor series about $\phi = \phi_0$, and got the area average diffusivity in the gap:

$$\left\langle \hat{D}_{\parallel} \right\rangle = \hat{D}_{\parallel} \big|\phi_0 + \frac{1}{2}\frac{\partial^2 \hat{D}_{\parallel}}{\partial \phi^2} \big|\phi_0 < (\Delta\phi^2) > + O(< (\Delta\phi^3) >) \tag{2.3}$$

They compared their results with diffusivity measured by Leighton and Acrivos [9] and found that their measurements were in close agreement at concentration of 40-50%. Krishnan and Leighton [34] then developed an experimental investigation into the dynamic viscous resuspension of bidisperse suspensions in an annular parallel plate. They used the model of Chapman and Leighton [13] to determine the effective diffusivity of the particulate phase in the plane of shear as a function of overall concentration as well as the relative size fraction of the large particles. They found that the effective diffusivity of a mixed suspension was somewhat less than the volume average of the diffusivities of monodisperse large and small sphere suspensions.

Krishnan, Beimfohr and Leighton continued their study on bidisperse suspensions. In their 1996's paper, they discussed experimental evidence for radial particle segregation in a parallel-plate geometry. The motion of colored tracer particles of a size different from the bulk suspensions was followed as a function of time. the tracer particles were seen to experience a constant drift velocity independent of their radial position. This was in addition to the random-walk motion arising from their interactions with other particles in the suspension. In the experiments of Abbott et al. [18], it was observed that larger particles migrated radially outward to region of lower shear stress in a wide-gap Couette device. In their experiments, large tracers were also observed to migrate radially outward, but in this case, the radial migration resulted in migration to regions of higher shear stress, contrary to expectations. They explained this apparent discrepancy in terms of a model that incorporated both the stress-induced migration and a curvature-induced migration flux as an opposing effect. They characterized the flux of tracer particles by two terms: the difference between the curvature and shear gradient migration β and the tracer diffusivity.

Miskin et al. [26] [27] did some researches on shear-induced migration of suspension in an inclined rectangular channel based on Leighton and Acrivos theory. At first the viscous suspension of identical, negative buoyant, spherical particles flowed down the rectangular channel driven only by gravity, i.e. no cross flow. It was shown that as the angle of inclination increased a constant concentration profile was reached within the resuspension layer and the velocity had its maximum value at the particle-clear fluid interface. Then a pressure-driven flow was added in a direction perpendicular to the original flow, i. e. a cross flow was presented. By increasing the flow rate sufficiently in y-direction at a fixed angle of inclination, it produced a value for the concentration within a layer close to the center of the channel that approached the limit maximum packing particle fraction. When the channel was inclined at larger angles to the horizontal, the ratio of the volume of clear fluid flowing in x-direction to that of y-direction at which the maximum packing condition was reached was found to decrease. When the initial feed concentration increased, the number of particle-particle interactions increased, and the value of k, a measure of the flow rate in the y-direction, at which the maximum packing condition was reached decreased.

In their following study they focused on the effect of two parameters γ (a/H) and β (dimensional scales for the length in x-direction is βH). It was found that the flow was very strongly influenced by γ. In particular, at small value of γ, the particle concentration within the suspension layer began to approach the maximum packing value and the thickness of this layer increased so as to cover almost the entire channel. This resulted in the formation of a thick, stagnant suspension layer above which a narrow layer of rapidly flowing clear fluid existed. As γ increased the height of the suspension layer decreased, as did the particle concentration within it. Additionally, an increase in γ resulted in an increase in the u-component of velocity within the suspension layer, but a decrease in the strong in the clear fluid layer. It was found that as β increased the particle concentration above the bottom surface of the channel began to decrease, and vice versa, but the thickness of the suspension layer was virtually unaffected. It was found that the particle concentration increased above the bottom surface of the channel and depleted from below the upper surface and the rate of increase or depletion depended on the γ.

Demonstrated by so many experiments, shear-induced migration was then theoretically modeled by several researchers. The main models are: Phillips' shear-induced diffusion model, Jenkins and McTigue', Nott and Brady's suspension balance model.

Ronald J. Phillips et al. [16] proposed a constitutive equation for computing the particle concentration and velocity fields in concentrated monomodal suspensions in 1991. The constitutive equation consisted of two parts: a Newtonian constitutive equation in which the viscosity depends on the local particle volume fraction and a diffusion equation that accounted for shear-induced particle migration.

For a given distribution of particle $\phi = \phi(x)$, the velocity and pressure fields could be found by solving the equations for linear momentum conservation and continuity.

$$\nabla \bullet \tau + \nabla p = 0 \tag{2.3}$$

$$\nabla \bullet v = 0 \tag{2.4}$$

where $\tau = -\mu(\phi)\dot{\gamma}$, and μ is the viscosity which can be expressed as

$$\mu = (1 - \frac{\phi}{\phi_m})^{-1,82} \tag{2.5}$$

These momentum and mass conservation equations should be combined with the particle diffusion equations to get the particle distribution and velocity fields.

Using a modified version of Leighton and Acrivos' flux expression, they developed the conservation equation for particles:

$$\frac{\partial \phi}{\partial t} = -\nabla \bullet (N_c + N_\mu + N_b) \tag{2.6}$$

where the left side is the rate of accumulation of particles, and N_c, N_μ and N_b are flux due to gradients in volume fraction, gradient in the viscosity, and Brownian diffusion of particles respectively.

$$N_c = -K_c a^2 (\phi^2 \nabla \dot{\gamma} + \phi \dot{\gamma} \nabla \phi) \tag{2.7}$$

$$N_\mu = -K_\mu \dot{\gamma} \varphi^2 (\frac{a^2}{\mu}) \frac{d\mu}{d\phi} \nabla \phi \tag{2.8}$$

K_c and K_μ are proportional constant of order unity that should be found from experimental data.

If the particle size is large enough, the flux due to Brownian diffusion could be negligible.

Phillips et al. solved the constitutive equations for one-dimension steady Couette and Poiseuille flow, and one-dimension transient start-up of steady shear flow in a Couette device. They compared their predictions for particle volume fraction and velocity fields with the NMR imaging, they found the predictions agreed remarkably well with the measurements.

In 1990, Jenkins and McTigue[23] proposed a theoretical model for the shear-induced migration in suspensions which was different from Phillips' model. In their model, they considered only the particle phase, and, in a manner akin to the treatment of dry granular materials. They proposed conservation equations for mass, momentum and fluctuation energy for the particular phase. Thus, in their one-component system, there was no diffusion equation, but the particle concentration, temperature (which was a measure of fluctuational motion of the particles), and bulk velocity fields were set up so as to satisfy the continuity, momentum and energy equations.

The Stokesian Dynamics was developed by Brady and Bossis in 1988. This molecular-dynamics-like approach could dynamically simulate the behavior of many particles suspended or dispersed in a fluid medium. This method exploits the fact that hydrodynamic interactions among particles can be decomposed into long-range mobility and short-range lubrication interactions. The long-range and the short-range lubrication interactions were calculated to form the grand resistance tensor. This grand resistance tensor related the vector of hydrodynamic forces and torques and stresses on the particles in a unit cell to their respective velocities and the rate of strain. Thus the particle velocity or forces could be computed at each time step.

Nott and Brady [24] developed another model using Stokesian Dynamics [25] to carry out the dynamic simulation of pressure-driven flow in a channel of a non-Brownian suspension at zero Reynolds number. Their model looked similar to Jenkins and McTigue's[23] model, but the physics underlining the model was quite different. Their simulations were performed for a range of ratio H/a and the bulk particle concentration ϕ_{bulk}. Starting from a homogeneous dispersion of particles in the channel, the simulation revealed that particles gradually migrated towards the center of the channel, resulting in a considerably inhomogeneous distribution and blunting, or flatting, of the particle velocity profile at steady state. They showed that the transit length scaled as H^3/a^2, which was much greater than the development length for a homogeneous Newtonian fluid. They found that the temperature was lowest at the channel center and increased until it reached a maximum at a distance of roughly two particle diameters from the wall, beyond which point it decreased and vanished at the walls. The temperature was a measure of the velocity fluctuations of particles about their local mean, and was closely related to the diffusivity –the diffusivity was properly thought of as proportional to the square root of particle temperature. Then they proposed a model for suspension flow, where there was no diffusion, but rather mass, momentum and energy balances written for the particulate phase in a manner analogous to molecular or atomic systems. The governing equations for the suspension are:

$$\frac{d}{dy}(\eta_s(\phi)\frac{d<u>}{dy}) = -1 \tag{2.9}$$

$$\frac{d}{dy}(p(\phi)T^{1/2}) = 0 \tag{2.10}$$

$$\eta_p(\phi)(\frac{d<u>}{dy})^2 - \alpha(\phi)T + \varepsilon^2\frac{d}{dy}(\kappa(\phi)\frac{dT}{dy}) = 0 \tag{2.11}$$

where, η is the viscosity of suspension, η_p is the relative viscosity of particle

α is the is the volume fraction dependent function

κ is the thermal conductivity

ε is a/R

With proper boundary conditions, the equation (2.9) to (2.11) could be solved for u, ϕ, and T.

They said that Phillips' model is "local" in a sense that the diffusivity is proportional to the local shear rate, while their model is non-local owing to the presence of the energy flux. They compared the prediction of these two models and found that for velocity profiles, the two models and simulations were in good agreement. For concentration profiles, the Phillips' model predicted a cusp with $\phi=\phi_m$ at the center, on the other hand, their model was almost identical to the simulation result, but their model displayed a sharp upturn to maximum packing in the thin boundary layers adjacent to the wall.

Many experiments and numerical calculation then developed to exam the accuracy of the theoretical model.

U. Schaflinger, A. Acrivos, and H. Stibi [14] performed experiments in a fully-developed Hagen-Poiseuille stratified channel flow with a clear fluid overlying a suspension to measure the pressure drop and the particle velocity at the suspension-clear fluid interface as a function of the well-mixed particle volume fraction ϕ, and a Shields number κ.

$$\kappa = \frac{9}{16}\frac{\nu_1 Q}{B^3 g\varepsilon} \qquad (2.12)$$

where $\varepsilon = \frac{\rho_2 - \rho_1}{\rho_1}$, B is the height of channel, Q is the volumetric flux of clear fluid per unit depth, ν_l is the kinematics viscosity of clear fluid. Thus κ is a measure of the strength of viscous forces relative to those of gravity.

Using Schaflinger et al.'s [16] analysis, they obtained the dimensionless pressure drop coefficient K in a 2-D Hagen-Poiseuille channel according to

$$\Delta P = P_{in} - P_{out} = KL\mu_1 Q/2B \qquad (2.13)$$

where L is the total length of the duct.

From the experiments, they observed that for a fixed feed concentration, the measured pressure drop coefficient K decreased abruptly at $\kappa \sim 0.006$ and attained values which were significantly lower than those predicted theoretically. At the same time, the interfacial waves were observed and eventually became very strong. A further increase in κ led to wave destruction and the appearance of clouds of detached particle moving relatively rapidly above the original interface. In this range the pressure drop coefficient increased and reached a value almost independent of κ. The intensity of wave breaking then lessened but remained significant. The measured particle velocity at the interface showed good agreement with theory for small values of κ. But at large values, the observed particle velocity at the interface was up to several times larger than that predicted due to the existence of a detached particle layer that moved very rapidly. Finally, they observed an additional flow instability when the bottom of channel was covered by a monolayer of particles.

Schaflinger, Acrivos and Zhang [29] investigated theoretically the viscous resuspension of negatively buoyant particles in a fully-developed 2-D Hagen-Poiseuille flow and in a corresponding plane, gravity-driven film flow. The theoretical approach was based on the model developed by Leighton and Acrivos [9]. It was shown in their results that, in a 2-D Hagen-Poiseuille flow, resuspension was restricted due to the existence of a plane with vanishing shear stress. Moreover, for large values of Shields number, a constant concentration within resuspended layer was reached, and the velocity had its maximum at the interface. For the film flow, the total thickness of the film was calculated as a function of the fraction of solid in a well-mixed suspension, with the angle of inclination and the relative density difference between the solids and the fluid as parameters. It was found that there existed a critical value of the fraction of solids above which the particles would settle out and form a sediment layer at the bottom that would continue to grow as a function of time. They also found that the experimental observations in a channel flow were in good agreement with the theory.

Chow, Sinton, Iwamiya, and Stephens[19] continued to use NMR imaging technique to serve on their study. They used the NMR technique to measure the evolution of suspension concentration profiles in Couette and parallel-plate flow devices upon shearing. In their study, neutrally buoyant suspensions of nearly monodisperse, Non-Brownian spherical particles at a volume fraction of 0.5 in a Newtonian fluid were used. The same flow cells and suspensions were also used in a rheometer to measure the changes in shear stress under identical experimental conditions such that a direct comparison can be made between the stress and concentration data. They observed that the shear stress increased at first and then decreased. The decrease in shear stress was indeed correlated to a decrease in the suspension concentration in the gap of Couette device. By mass balance, any depletion in particles in the annulus must be compensated by the gain in particles in the stagnant region beneath the bob, as was seen in NMRI data. These results were direct verification of shear-induced particle migration normal to the plane of shear, as described by Leighton and Acrivos [9]. The increase in the shear stress at low stain was likely to be related to the redistribution of particle concentration in the shear plane. In torsional flow between parallel plates, the concentration profiles measured by NMRI were nearly uniform with no evidence of bulk particle migration in the radial direction, but some decrease in the apparent viscosity was observed. The constitutive model by Phillips et al. [16] would predict inward particle migration in parallel plates, that was in disagreement with experimental evidence. They thought this was an important clue in developing or evaluating more general description of particle migration.

Hampton et al. [20] focused their study of shear-induced migration on the demixing of neutrally buoyant suspension of spheres during slow, pressure driven flows in circular conduits. They measured the distribution of the solid fraction of particle and the suspension velocity at different lengths from a static in-line mixer. The experiments were conducted over a range of volume average solids fraction ($0.1 \leq \phi \leq 0.5$), and at two different ratios of the particle radius to the radius of the circular conduit ($a/R=0.0256$, $a/R=0.0625$). At $\phi \geq 0.20$, the particles rapidly migrated to low-shear-rate region in the center of the conduit. This migration results in a blunting of the velocity profile, relative to the parabolic profile observed in homogeneous Newtonian fluids. For the flow geometry with the smaller ratio a/R, the ϕ profile builds to a sharp maximum or cusp in center. Particle structures are observed in the experiments with higher a/R ratios. The entrance length for development of ϕ and v fields are strong functions of a/R ratio and ϕ_{bulk}. Entrance length decreased rapidly as ϕ and a/R increased. Over the range of their data, the v profiles were found to develop more rapidly than the ϕ profiles. They compared their experimental results with fully developed flow predictions from several theoretical models. They found that shear-induced migration model established by Phillips et al. [16] more accurately predicted the experimental results at smaller a/R ratio, while at large a/R ration, the suspension balance model developed by Jenkins and McTigue [31], and Nott and Brady [24] predicted some qualitative features of experimental results better

Mondy et al. [21] conducted a study to determine the effect of particle shape on the shear-induced migration in 1993. They measured the migration of rods suspended in Newtonian fluids and studied the behavior of several suspensions. Each suspension contained well-characterized uniform rods with aspect ration ranging from 2 to 18 at either 0.30 or 0.40 volume fraction. At the same volume fraction of solids, the steady-state, radial concentration profiles for rods were independent of aspect ratio and were indistinguishable from those obtained from suspended spheres and only minor differences near the wall appeared to differentiate the profiles. Data taken during the transition from a well-mixed suspension to final steady state showed that the rate of migration increased as the volume of the individual rods increased. They also compared their results with Phillips' model [16] and found some difference between the experimental results and prediction.

While most of experimental studies of shear-induced migration use the NMR imaging technique, Koh et al. [22] developed a new technique to work on it. They utilized the well-known laser-Doppler anemometry technique to measure the velocity and concentration profiles in concentrated suspension flows in a rectangular channel. To circumvent the problem of optical turbidity, the refractive indices of the solid and liquid were closely matched. The residual turbidity, owing to small mismatches of the refractive indices, as well as impurities in the particles, allowed a Doppler signal to be detected when a particle passed through the scattering volume. By counting the number of Doppler signals in a period of time, the local volume fraction was measured. In their study, they performed experiments for a range of values of the various important dimensionless parameter: bulk particle (ϕ_{bulk}), the particle size to gap ratio (a/R), and the particle Reynolds number. They found that the flow of concentrated suspension in a rectangular channel was characterized by a blunting of particle velocity profile. The magnitude of this blunting increased with increase of either the bulk particle concentration of the suspension or the ratio of the particle size to the gap width of the channel. They also found that local particle concentration was not uniform across the gap of the flow channel; in general, the particle tended to concentrate near the center of the flow channel. The concentration profiles for ϕ_{bulk} =0.2 and 0.3 were independent of particle size. One of most interesting phenomena inferred from this study was that there was a relatively large slip velocity between the particle and the fluid, which increased as the bulk particle concentration increased.

Averbakh et al. [37] also employed Laser-Doppler technique to study liquid-solid concentration suspensions. Matching of refractive indices of the continuous and dispersed phases was used to establish optimal conditions for the quality of the optical signals and can be used to measure concentration in the flowing dispersion. Laser-Doppler anemometry was applied to measure velocity profiles in a rectangular duct and to detect velocity fluctuations in the viscous flow of the concentrated suspension induced by the particle presence. The existence of a net drift of particles in a concentrated suspension was demonstrated. They found from their experiments that for vertical velocity profiles, they were curved rather than flat along the center portion at ϕ=0.45, and for horizontal velocity profiles under same volume fraction, they were different from those obtained for the less concentrated suspensions. It was readily seen that horizontal velocities prevailed and were directed from side wall toward the center. It was evident that these velocities increased in magnitude as the maximum vertical velocities increase. Since the fully developed vertical velocity profile was almost instantaneously established it was concluded that the measured horizontal velocity indicated the existence of a net particulate drift from the walls to the center.

Nir and Acrivos [30] reported their study on sedimentation and sediment flow on inclined surfaces. Also based on Philips model, they analyzed the steady sedimentation by considering the combined effects of settling hindrance, bulk motion and particle resuspension. The coupled momentum and mass balances suggested that a thin high-density sediment layer would form over the inclined surface, reminiscent of the thin thermal boundary layers in the classical problem of free convection. Using the Karman-Pohlhausen integral method for approximation velocity and concentration, they obtained the result that for a given value of the particle volume fraction in the unsettled suspension, a steady flow of the sediment could be maintained only if the angle of inclination exceeded a minimum value. Their analysis also predicted the existence of a sharp discontinuity in the particle volume fraction across the suspension -- sediment interface along which the bulk velocity had a local maximum. High particle volume fraction within the sediment was predicted when the unsettled suspension was either very dilute or very concentrated. This led to the formation of relatively large sediment layer thickness which reflected the fact that a large body force was required in these two limiting cases to overcome the viscous resistance to flow near the inclined boundary.

Zhang and Acrivos [2] published their paper in 1993. They investigated the viscous resuspension of heavy particles in fully developed laminar flows in a horizontal pipe theoretically by extending Phillips' model to cases in which all three velocity components were non-zero due to the existence of a secondary motion within the cross section of the pipe. They employed the Galerkin finite-element method to solve the resulting mathematical system equations. Compared with analytical solution, they found that their numerical solution curves almost coincided with the analytical profile, except near the centerline of the pipe. They also presented their predictions for the flow of suspension containing heavy particles and compared them to the experimental results reported by

Altobelli et al. [17]. They found that under action of gravity, a region of clear fluid appeared at the top of pipe in most cases, and that this region became smaller when the flow rate Q and/or the mean concentration ϕ_m increased, illustrating the enhancement of the resuspension with increasing of Q and ϕ_m. They also saw the axial velocity profiles were blunt, especially when the mean concentration ϕ_m was high, and the location of maximum axial velocity shifted above centerline. Their results were also capable of predicting the experimental observation [17] that, when the mean concentration was high, the particle formed a highly concentrated core, centered slightly above the pipe centerline and near the location of maximum axial velocity. When the flow rate increased, the core became bigger and the corresponding maximum value of the concentration within the core became slightly larger. They explained the mechanism causing the formation of the core was that, when the particle concentration was high, the diffusion due to the shear-rate gradient was strong and created a large upward flux of particles. On the other hand, when this part of the shear-induced diffusion was not strong enough to support the concentrated core, as in the case of low mean concentrations, the concentration deceased monotonically from bottom to the top of the pipe.

Pelekasis and Acrivos [31] examined the steady laminar flow of a well-mixed suspension of monodisperse solid spheres convected steadily past a horizontal flat plate and sediment under the action of gravity in 1995. They found that in the limit as $Re \rightarrow \infty$ and $\varepsilon \rightarrow 0$, where Re was the bulk Reynolds number and ε was the ratio of particle radius α to the characteristic length scale L, the analysis of determining the particle concentration profile had several aspects in common with that of obtaining the temperature profile in forced convection heat transfer from a wall to a fluid stream moving at high Reynolds and Prandtl number. Specially, it was found that the particle concentration remained uniform throughout the O($\mathrm{Re}^{-1/2}$) thick Blasius boundary layer except for two O($\varepsilon^{2/3}$) thin regions on either side of the plate, where the concentration profile became non-uniform owing to the presence of shear-induced particle diffusion. The system of equations within this concentration boundary layer admitted a similarity solution near the leading edge of the plate, according to which the particle concentration along the top surface of the plate increased from its value in the free stream by an amount proportional to $X^{5/6}$, with X measuring the distance along the plate, and decreased in a similar fashion along the underside. Unlike the case of gravity settling on an inclined plate in the absence of a bulk flow at infinity considered by Nir and Acrivos [30], here the concentration profile remained continuous everywhere. It was also found that, as predicted from the similarity solution, there existed a value of X at which the particle concentration along top side of plate attained its maximum value and that, beyond this point, a stagnant sediment layer would form that grows steadily. But again in conformity with the similarity solution, for values of X sufficiently far from the leading edge along the underside of the plate, a particle free region was predicted to form adjacent plate.

Fang and Phan-Thien [28]conducted their study on the numerical modeling of particle migration in the concentrated suspension by finite volume method. Based on Phillips' model, they extended Phillips' study of one-dimension to two-dimension flows with arbitrary geometry and boundary conditions. The numerical implementation was benchmarked against the exact solution in circular Couette flow, including the transient case where a closed-form solution was not available. For the steady state, the numerical results were indistinguishable from the analytic solution, and for transient flow, the numerical results were also in excellent agreement with the experimental data. Then they give the numerical predictions in an eccentric circular geometry, where the inner cylinder was placed off the axis of the outer cylinder. The results showed that particles migrated towards the outer cylinder, but at different rates depending on their azimuthal position. At large eccentricity ratios, they found a recirculation in the wide gap. The magnitude of the recirculation was very small, with a corresponding low shear rate. Consequently, a local high volume fraction, near to the maximum volume fraction, was built up in this region leading to a very strong coupling between the kinetics and the volume fraction: a small change in the volume fraction led to a significant change in the kinematics.

Phan-Thien et al. [33] reported on experimental measurements and numerical predictions of shear-induced migration of particles in concentrated suspensions subjected to flow in the wide gap between a rotating inner cylinder placed eccentrically within a fixed outer cylinder. They used the NMR imaging technique to measure the time evolution of concentration and velocity. They found that depending on geometry, the final region of high solids concentration could either be at the outer wall or in a region within the gap. Then they employed the explicit pseudo-transient finite volume code to solve the governing equations developed by Phillips et al. They compared the numerical results with the experimental results and found that there was acceptable agreement. But some important qualitative features of the flow could not be reproduced by the numerical model. Specifically, the flow reversal was not seen in the 0.5 eccentricity ratio cases.

Phan-Thien and Fang [32] investigated the flow behavior of Phillips et al's suspension model in channel and Pulsatile flows by a combination of analytical and numerical methods (unstructured finite volume method). In channel flow, they found that the kinematics developed faster than the concentration, with an entrance length of $0.2H^3/\alpha^2$, where h is the channel half width, and α is the particle radius. The corresponding entrance length of the concentration was $0.5H^3/\alpha^2$. In time-periodic flows, the steady-state kinematics and stresses were periodic in time, and yet the steady-state concentration (and thus the viscosity) was time-dependent.

Subia et al. [36] utilized a Galarkin finite element Navier-Stokes solver to simulate the complex behavior of suspensions containing neutrally-buoyant spheres in a Newtonian fluid. Based on Phillips et al's [16] constitutive equation. They studied the shear-induced migration problems numerically. The two-dimensional and three-dimensional (axisymmetric) model was benchmarked with a variety of single-phase and two-phase analytic solutions and experimental results. Then they described new experimental results using NMR imaging to determine non-invasively the evolution of the solids-concentration profiles of initially well-mixed suspensions subjected to slow flows such as flow between counter-rotating eccentric cylinders and piston-driven flow in a pipe. It showed good qualitative and quantitative agreement of the numerical predictions with the experimental measurements.

In summary, shear-induced migration in suspensions under shear flow was demonstrated by many experimental studies in all kinds of flow channels, such as Gadala-Maria and Acrivos [8] first observed in their experiment. The theoretical models for this shear-induced migration were developed by several authors. They modeled the suspensions as an effective continuum. Based on Acrivos and Leighton's analysis [9][10], Phillips et al. [16] established the shear-induced migration model, and developed a constitutive equation for the suspension. Jenkins and McTigue [23], Nott and

Bradyc[24], proposed another model, say, suspension balance model which provided a non-local description of suspension behavior in terms of the particle's velocity fluctuation. Several numerical studies were finished based on these models, and provided modifications to the original models.

In our study, we will incorporate the modified Phillips et al's diffusive flux model into a finite element method formulation to model developing shear-induced migration in pipe flow of suspension.

CHAPTER 3

THEORETICAL MODEL

The main interest of this chapter is to evaluate the assumptions based on the considerations of the problem of viscous suspension flow, to discuss the technique of shear-induced migration modeling, and to present the derivation of the governing equations of the viscous suspension.

3.1 Basic Assumptions and Flow Analysis

This analysis is carried out under the following assumptions and considerations.

1. Incompressible flow.
2. Uniform inlet velocity profile.
3. Laminar flow region.
4. The suspension flow is considered as a single continuum.
5. Thermally insulated channel walls without heat transfer.
6. Negligible deposited particle layer thickness.
7. Single size sphere particle.
8. Under the effects of particle momentum, gravity and viscosity forces.
9. Negligible Brownian diffusivity effect.
10. Uniform particle concentration at the inlet of flow channel.
11. Non slip fluid flow condition at the channel walls.

In this work the suspension is in air atmospheric pressure and normal room temperature of 20°C, with a steady free stream flow velocity. The flow is a low Reynolds number flow and the velocity is small. It is therefor reasonable to consider this flow as incompressible and isotropic. The Reynolds number is defined as $Re = \rho_f UL/\mu_f$. where, L and U are characteristic length and velocity respectively, ρ_f is the density of suspending fluid, and μ_f is the dynamic viscosity of suspending fluid. The Reynolds number in this study is order of one. At low Reynolds number the flow is laminar and the inertial effect may be ignored.

As we mentioned in chapter two, there are two major theoretical models for the shear-induced migration in viscous suspensions. One is Phillips' shear-induced migration model [16], and the other is Nott and Brady's suspension balance model [24]. Both models consider the viscous suspension as a continuum Newtonian fluid with effective properties dependent on concentration.

In this study there is no heat transfer between the channel walls and the flow. The deposition and resuspension of particles have no effects on the temperature of the flow field and channel walls.

The Brownian diffusion force is a major effect in the concentration distribution of the suspension. But the diameter of particles in this study is bigger than 0.762mm. At this size of particle, the diffusivity is very small and this effect can be neglicted.

It is assumed that the particulate phase can be represented by a single mean size of spherical shape so that all particles are considered to have the same diameter.

Although there is ample evidence for existence in suspension flows of a relative velocity along the solid walls, calculations for the fully developed flow in a channel [29] showed that the addition of a relative velocity had a limited effect on the pressure drop and the height of the resuspension region for any given set of parameters. Based on these calculation, we consider that there is no slip at the channel walls.

In our analysis, we will utilize the theoretical model developed by Phillips et al to study the behavior of developing viscous suspension flow in a two dimensional channel. The governing equations of viscous suspension based on the Phillips' model are described in the following sections.

3.2 Development of the Governing Equations

A. Effective properties

The effective physical properties are the bulk-average effective properties. They are functions of the local volume fraction of the particles. The effective density can be written as:

$$\rho=\rho_f(1+\varepsilon\phi) \tag{3.1}$$

where

$$\varepsilon=\frac{\rho_p-\rho_f}{\rho_f}\geq 0 \tag{3.2}$$

ρ_f is the density of pure fluid, ρ_p is the material density of the particle, and ϕ is the local volume fraction of particles.

There are two expressions of effective viscosity μ given in the literature.

Krieger(1972) gave

$$\mu=\mu_f(1-\frac{\phi}{\phi_m})^{-1.82} \tag{3.3}$$

Leighton & Acrivos [10] gave

$$\mu=\mu_f\left(1+\frac{1.5\phi}{1-\frac{\phi}{\phi_m}}\right)^2 \tag{3.4}$$

where μ_f is the viscosity of pure fluid, ϕ_m is the critical particle volume fraction beyond which the suspension can not flow. If we use the first expression (3.3) to calculate the effective viscosity, ϕ_m is 0.68 [16]. If we use the second expression (3.4), ϕ_m is 0.58 [23]. In our study, we employ the second expression (3.4) to calculate the effective viscosity. Thus the value of ϕ_m is set to 0.58 in this study.

B. Momentum conservation and continuity equations

As we mentioned before, both theoretical models (Phillips model and Brady and Nott model) consider the suspension flow as a Newtonian continuum fluid. We can give the momentum conservation and continuity equation for the suspension flow with effective properties as following.

$$\rho(\phi)\frac{D\vec{u}}{dt} = \nabla \bullet \sigma + (\rho(\phi) - \rho_f)g \tag{3.5}$$

$$\nabla \bullet \vec{u} = 0 \tag{3.6}$$

where $$\sigma = -pI + \mu(\phi)(\nabla u + \nabla u^T) \tag{3.7}$$

U, σ and P are the bulk velocity, stress and pressure field respectively, and $(\rho(\phi)-\rho_f)$ is the effective body force, The effective viscosity $\mu(\phi)$ is given by (3.4).

C. Development of the diffusion equation

As mentioned in Chapter two, Acrivos and Leighton [9] [10] explained that the shear-induced migration is due to irreversible interparticle interactions. Although the interparticle interactions also include three-body interactions, it is found that the measurements of the diffusion flux induced by this three-body collisions are approximately an order-of –magnitude lower than those induced by two-body interactions. Thus the development of diffusion equation is based entirely upon the effects of two-body, irreversible interactions.

Particle fluxes that result from two-body interactions are explained in the following two sections.

1. Effect of spatially varying interaction frequency

Figure 3.1 shows a schematic diagram of irreversible two-body collisions. A collision shown in fig. 3.1 (a) occurs when two particles embedded in adjacent shearing surfaces move past one another. Since these collisions can cause a particle to be irreversibly moved from its original streamline, a particle which experience a higher frequency of collisions from one direction than from opposing direction will migrate normal to its shearing surfaces and in the direction of lower collision frequency.

The number of collisions experienced by a test particle will scale as $\dot{\gamma}^{2}\phi$ [10], where $\dot{\gamma}$ is the magnitude of local shear rate, and is proportional to the square root of the second invariant of the rate of the deformation tensor d,

$$\dot{\gamma} = (2d : d)^{\frac{1}{2}} \tag{3.8}$$

where $d \equiv \frac{1}{2}(\nabla u + \nabla u^{T})$.

The variation in the collision frequency over a distance of O(a) is then given by $a\nabla(\dot{\gamma}\phi)$. Assuming the particle migration velocity is linear by proportional to this variation in the collision frequency and each of these two-body interaction gives rise to a displacement of O(a) leads to an expression for the flux N$_c$ caused by this effect as

$$N_{c} = -K_{c}a^{2}(\phi^{2}\nabla\dot{\gamma} + \phi\dot{\gamma}\nabla\phi) \tag{3.9}$$

where K_c is a proportional constant of order unity that can be determined from experimental data.

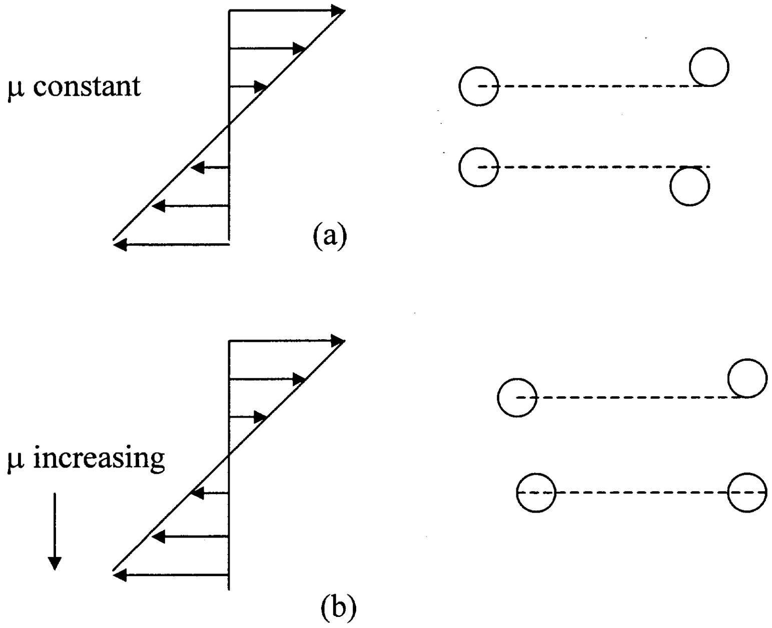


Figure 3.1(a)(b) Schematic diagram of irreversible two-body collision.

The first term on the right side of Eq. (3.8) implies that even with no gradient in the particle volume fraction, migration will result if the particles on one side of a test particle are moving past it more rapidly than on the other side. Because this variation gives a higher number of interparticle interactions on the side with higher $\dot{\gamma}$. The second term on the right side of Eq. (3.8) states that a gradient in particle volume fraction will cause a spatial variation in the frequency if interactions.

2. Effect of spatially varying viscosity

There exists another interaction between two particles that is affected by the spatially varying viscosity $\mu(\phi)$ caused by the gradients in the particle concentration. This effect is shown on fig. 3.1(a) and 3.1(b). A gradient in viscosity results in the resistance to motion experienced by one of the interacting spheres being greater than that experienced by the other. This difference in resistance causes the center of rotation for the doublet to be displaced from its position in the absence of a viscosity gradient. For spherical particle, it moves from the midpoint of the line joining the sphere centers toward the direction of higher viscosity. The magnitude of this displacement was given quantitatively by Leighton and Acrivos [10]. They assumed the magnitude of this displacement was proportional to $(a/\mu)\nabla\mu$, multiplied by the particle radius. For small gradient in concentration, the variation in viscosity is linear in the concentration gradient.

The above expression, multiplied by the rate of interactionns $\dot{\gamma}\phi$, gives the corresponding drift velocity. Finally, multiplied by ϕ and using Eq. (3.3) to express the gradient of viscosity in terms of $\nabla\phi$ gives the flux

$$N_\mu = -K_\mu \dot{\gamma} \phi^2 \left(\frac{a^2}{\mu}\right) \frac{d\mu}{d\phi} \nabla\phi \tag{3.10}$$

where N_μ is the particle flux due to the spatial variation in the viscosity, and K_μ is the constant of order unity that can be determined by experiment.

In this expression, the shear-induced migration rate is independent of the viscosity of the suspending fluid as demonstrated by Abbott et al [18].

3. Conservation equation for particles

In this analysis, we employ a cartesian coordinate system using t for time, x_1 for horizontal coordinates, and x_2 for vertical coordinates. The bottom wall of the channel is at $x_2 = 0$ and the top wall at $x_2 = h$, where h is the channel height.

The particle conservation equation can be obtained by Fick's law.

$$u \bullet \nabla \phi = -\nabla \bullet J \qquad (3.11)$$

where J is the total flux, which includes the shear-induced diffusion flux N_D and gravitational sedimentation flux N_G in this analysis.

$$J = N_D + N_G \qquad (3.12)$$

The Stokes settling velocity of a sphere in the suspension can be expressed as

$$u_{settle} = \frac{2}{9}\frac{ga^2 \Delta\rho}{\mu_f} \qquad (3.13)$$

where $\Delta\rho = \rho_p - \rho_f$,

In suspension flow, there is a relative velocity between the particle and bulk flow. This velocity can be expressed as

$$u^* = u_{settle} f(\phi) \qquad (3.14)$$

where f(ϕ) is the so-called hindrance function [2].

$$f \approx \frac{1-\phi}{\mu} \qquad (3.15)$$

So the expression for the gravitational sedimentation flux becomes

$$N_G = \frac{2}{9}\frac{a^2(\rho_p - \rho_f)}{\mu_f}\phi f(\phi) g \qquad (3.16)$$

and $N_D = N_c + N_\mu = -K_c a^2 (\phi^2 \nabla \dot{\gamma} + \phi \dot{\gamma} \nabla \phi) - K_\mu \dot{\gamma} \phi^2 (\frac{a^2}{\mu}) \frac{d\mu}{d\phi} \nabla \phi$

$$= -a^2 \dot{\gamma} (K_c \phi + K_\mu \phi^2 \frac{1}{\mu}\frac{d\mu}{d\phi}) \nabla \phi - a^2 K_c \phi^2 \nabla \dot{\gamma}$$

$$= -D_c \nabla \phi - D_s \nabla \dot{\gamma} \qquad (3.17)$$

where N_c is the flux caused by effect of spatially varying interaction frequency, and N_μ is the flux due to the spatial variation in the viscosity.

$$\mathrm{D_c} = a^2 \dot{\gamma}(K_c\phi + K_\mu\phi^2 \frac{1}{\mu}\frac{d\mu}{d\phi}), \text{ and}$$

$$\mathrm{D_s} = a^2 K_c \phi^2$$

Combining Eq. (3.11) to (3.17) results in the diffusion equation for the particle volume fraction.

$$\frac{D\phi}{Dt} = \nabla\cdot(-a^2 \dot{\gamma}(K_c\phi + K_\mu\phi^2 \frac{1}{\mu}\frac{d\mu}{d\phi})\nabla\phi - a^2 K_c\phi^2\nabla\dot{\gamma}$$

$$+\frac{2}{9}\frac{a^2(\rho_p - \rho_f)}{\mu_f}\phi f(\phi) g) \qquad (3.18)$$

Eq. (3.5), (3.6) and (3.18) are the governing equations for a highly concentrated suspension of monomodal spheres undergoing nonuniform shear flows.

3.3 The Dimensionless Form of Governing Equations

The dimensionless parameters are as follows

$X_i = x_i / h$,

$U_i = u_i / u_0$,

$T = \frac{t}{\frac{h}{u_0}}$,

Reynolds Number, Re$= \frac{\rho_f u_0 h}{\mu_f}$,

Modified Froude Number, Fr$= \frac{u_0 \mu_f}{(\rho_p - \rho_f) h^2 g}$, The relative density ratio of the suspended particles to the suspending fluid,

$$\varepsilon = \frac{\rho_p - \rho_f}{\rho_f},$$

And the square of the ratio of the particle diameter α to the characteristic length scale,

$$\lambda = (\frac{a}{h})^2 .$$

where h is the characteristic length, the height of the channel, and u_0 is the characteristic velocity, the inlet uniform velocity.

The dimensionless forms for the coefficients Dc and Ds based on Acrivos and Leighton's theory [2] are

$$\hat{D}_C = \frac{D_C}{a^2 \dot{\gamma}} = \frac{1}{3}\phi^2 (1 + \frac{1}{2} e^{8.8\phi}) + 0.6\phi^2 \frac{1}{\mu}\frac{d\mu}{d\phi} \tag{3.19}$$

$$\hat{D}_s = \frac{D_S}{a^2} = 0.6\phi^2 \tag{3.20}$$

with Eq. (3.4) for effective viscosity μ.

By using the channel height h as the characteristic length and the initial uniform velocity u_0 as the characteristic velocity, we can write the governing equations in dimensionless form.

$$\frac{\partial U_j}{\partial X_j} = 0 \tag{3.21}$$

$$\mathrm{Re}(1 + \varepsilon\phi)(\frac{\partial U_i}{\partial T} + U_j \frac{\partial U_i}{\partial X_j}) = \frac{\partial}{\partial X_j}[-P\delta_{ij} + 2\mu(\phi) d_{ij}] - \frac{1}{Fr}\phi\delta_{i2}$$

$$i=1,2 \tag{3.22}$$

$$\frac{\partial \phi}{\partial T} + U_j \frac{\partial \phi}{\partial X_j} = \lambda \frac{\partial}{\partial X_j}[\hat{D}_C(\phi)\dot{\Gamma}\frac{\partial \phi}{\partial X_j} + \hat{D}_S(\phi)\frac{\partial \dot{\Gamma}}{\partial X_j} + \frac{2}{9\,Fr}\phi f(\phi)\delta_{j2}] \quad (3.23)$$

where summation convention for repeated indices j=1,2 is used.

d_{ij} is the deformation tensor, which is expressed as $d \equiv \frac{1}{2}(\nabla u + \nabla u^T)$.

δ_{ij} is the Kronecker delta.

3.4 Boundary and Initial Conditions

To complete the formulation, we need the boundary and initial conditions.

At the inlet of the channel, we have the uniform velocity.

$U_1 = u_0$, $U_2 = 0$, at $X_1 = 0$.

As mentioned in 3.1, we assume there is no slip for the bulk at the channel wall,

$U_i = 0$ at $X_2 = 0$ and $X_2 = 1$ for i=1,2.

A zero normal stress is specified at the channel exit.

Boundary conditions on particle volume fraction include a constant value maintained at the inlet

$\phi = \phi_0 =$ constant at $X_1=0$.

Zero diffusive flux conditions on particle concentration at both walls and exit.

$(N_G + N_D) = 0$ at $X_2 = 0$, $X_2 = 1$, and $X_1 = L$,

where L is the length of the channel.

CHAPTER 4

NUMERICAL SIMULATION

In this chapter we will discuss the steps and methods that are needed in obtaining the solution to the concentration and velocity profiles in the viscous suspension in a two-dimensional parallel-plate channel. We may divide the process into several steps. The first step is to derive the weak form for each of the governing equations. The second step is to replace the pressure term in the weak form by velocity terms using the penalty method. The third step is to develop a program to solve the coupled governing equations.

4.1 Finite Element Modeling

4.1.1 Introduction to Finite Element Method

To obtain a solution to the governing equations of the viscous suspension flow by analytical method is a formidable task and unrealistic. In this case, we will use the approximate method instead. The finite element method and the variational methods are most frequently used in literature [38].

In the solution of a differential equation by a variational method, the equation is transformed into an equivalent weighted-integral form and then the approximate solution over the domain is assumed to be a linear combination ($\sum_j c_j \phi_j$) of appropriately chosen approximation functions ϕ_j and undetermined coefficients c_j. The coefficients c_j can be determined as the integral equivalent to the original differential equation is satisfied. The finite element method can be viewed as an element-wise application of the variational method. The method is endowed with three basic features. First, a geometrically complex domain of the problem is presented as a collection of geometrically simple subdomains, say finite elements. Second, over each finite element, the approximation functions are derived using the basic idea that any continuous function can be represented by a linear combination of algebraic polynomials. Third, algebraic relations among the undetermined coefficients are obtained by satisfying the governing equations, often in a weighted-integral sense.

4.1.2 Integral Formulation and Weak Form

In finite element method, an integral statement is used to develop algebraic relations among the coefficients u_i of the approximation

$$u \approx U_N = \sum_{j=1}^{n} u_j \phi_j \tag{4.1}$$

where u represents the solution of the differential equation, A(u)=f in Ω and ϕ_j are functions preselected such that the specified boundary conditions of the problem are satisfied. We can require the approximate solution U to satisfy the differential equation in the weighted-integral sense, for a domain of Ω,

$$\int_{\Omega} wR dx = 0 \tag{4.2}$$

where R is the residual, A (U_N) - f = R , and w is the weight function.

From (4.2), we obtain as many linearly independent equations as there are independent functions for w. Thus we can obtain as many algebraic equations as there are unknown coefficients in the approximation. A weak form is a weighted-integral statement of a differential equation in which the differentiation is distributed among the dependent variable and the weight function and includes the natural boundary conditions of the problem.

4.1.3 Weak Formulation for the Governing Equation

The system of governing equation system in our study is more complicated than the equations for the pure fluid. The diffusion coefficient Dc depends on both the scalar Φ and shear rate $\dot{\gamma}$. Furthermore, in the equation of flux balance, Eq. (3.18) there is a third-order derivatives of the flow rate, i.e. the second term on the right hand side of this equation. and the viscosity and diffusion coefficients are all highly non-linear functions of Φ. To solve this system of equations, we have to linearize all non-linear terms.

There are three steps in the development of the weak form of any differential equation and boundary conditions.

A. Move all expressions of the differential equation to one side, and multiply the entire equation with a weight function w.

B. Use integration by parts to distribute differentiation evenly between the dependent variable and the weight function, and use the boundary terms to identify the form of primary and secondary variable.

C. Modify the boundary terms by restricting the weight function to satisfy the homogeneous form of the specified essential boundary conditions and replacing the secondary variables as their specified values.

Follow these steps, we will formulate the weak form for momentum equation (3.22), using ω as the weight function.

$$\int_{\Omega}\omega(1+\varepsilon\phi)(\frac{\partial U_i}{\partial t}+U_j\frac{\partial U_i}{\partial X_j})d\Omega+\frac{1}{\mathrm{Re}}\int_{\Omega}\mu\frac{\partial\omega}{\partial X_j}(\frac{\partial U_i}{\partial X_j}+\frac{\partial U_j}{\partial X_i})d\Omega-\int_{\Omega}\mu\frac{\partial\omega}{\partial X_i}\frac{\partial U_j}{\partial X_j}d\Omega$$

$$=-\frac{1}{\mathrm{Re}\ Fr}\int_{\Omega}\omega\phi\delta_{i2}d\Omega \qquad (4.3)$$

and the weak form for flux balance equation (3.33), where ϖ is the weight function..

$$\int_{\Omega}\varpi(\frac{\partial\phi}{\partial t}+U_j\frac{\partial\phi}{\partial X_j})d\Omega+\lambda\int_{\Omega}\frac{\partial\varpi}{\partial X_j}[\hat{D}_C(\phi)\dot{\gamma}\frac{\partial\phi}{\partial X_j}+\hat{D}_s(\phi)\frac{\partial\dot{\gamma}}{\partial X_j}$$

$$+\frac{2}{9\,Fr}\phi f(\phi)\delta_{ij}]d\Omega=0 \qquad (4.4)$$

The pressure in the momentum equation is expressed by penalty method, so the pressure is no longer a primary field variable. We will explain the penalty method in the next section.

In our study, we employ isoparameter elements and the weighting functions ω and ϖ are the same as shape functions φ.

Moreover, we use the isoparametric, quadrilateral, Lagrangian elements, thus, the dependent variables in Eq. (4.3) and (4.4) were interpolated in terms of the quadratic shape functions φ_k,

$$U_i=\varphi_k U_{ik},\qquad \phi=\varphi_k\Phi_k,\qquad \mathrm{i}=1,2, \qquad (4.5)$$

where U_{ik} and Φ_k denote the values of the velocity components and concentration at the nodal point k respectively.

4.2 Penalty Formulation

In the penalty method [38] [39] the continuity equation is perturbed with a small parameter times the pressure. Next the pressure is eliminated, thus uncoupling the momentum equation and the continuity equation. The main advantage of the penalty method is the large reduction of the system of equations as well as the fact that partial pivoting is no longer necessary.

The idea of the penalty method is to perturb the continuity equation with a small term containing the pressure:

$$\varepsilon p + div \;\; u = 0 \tag{4.6}$$

with ε a small parameter. We can consider this perturbation as the introduction of a slight artificial compressibility. Replacing the continuity equation by relation (4.6) results into

$$p = -\tau div \;\; u \tag{4.7}$$

where τ is the penalty parameter, which is a very large positive constant..

From Eq. (4.7) it follows that

$$\int_{\Omega} p d\Omega = \int - \tau div \; u \; d\Omega \tag{4.8}$$

it means that after u has been computed, the additive constant for the pressure is determined by Eq. (4.8).

The pressure in the momentum equation in our study can be written in dimensionless form as:

$$P = -\tau \operatorname{Re} \;\; \mu \frac{\partial U_j}{\partial X_j} \tag{4.9}$$

where P=$\dfrac{p}{\rho_f u_0^{\,2}}$.

Substituting the penalty expression for pressure into the momentum equation we get the weak form for the momentum equation as Eq. (4.3)

4.3 Linearization Technique for System of Non-Linear Equation

There are several methods to linearize systems of non-linear equations. In this way the solution of the non-linear equations is reduced to the solution of a sequence of systems of linear equation.

The system of non-linear equations can be written formally as

$$K(u)u = f \tag{4.10}$$

Picard iteration, also called successive substitution or direct method, is characterized by the fact that one of the vector u in Eq. (4.10) is taken on a preceding iteration level, whereas the other one is taken at the new level. So a Picard iteration could be

a. Start: choose u^0.

b. For I=1,2,... Solve:

   $K(u^{i-1})\, u^i = f$,

   or alternatively

   for I=1,2, ... solve:

   $K(u^i)\, u^{i-1} = f$.

In our study, the convection terms in Eq. (4.3) and (4.4) are non-linear terms, and they can be linearlized as

$$U_j \frac{\partial U_i}{\partial x_j} = U_j^{k-1} \frac{\partial U_i^k}{\partial x_j},$$

and

$$U_j \frac{\partial \Phi}{\partial x_j} = U_j^{k-1} \frac{\partial \Phi^k}{\partial x_j}.$$

Picard method is simple to program and have in general a large region of convergence. Though its rate of convergence is slow for large Reynolds number, we can employ this method for our low Reynolds number problem.

4.4 Numerical Integration Formulas

As we mentioned in section 4.1, we use the quadrilateral Lagrangian elements in our study. Here we will discuss the method to calculate the resulting integrals in the weak form of governing equations.

We use a second-order Gaussian quadrature to calculate the integrals, but we use a first-order quadrature to evaluating the penalty term in Eq. (4.3) in order to maintain numerical stability.

Quadrature formulae for integrals defined over a rectangular master element can be written as

$$\int_{\Omega} F(\xi,\eta) d\xi d\eta \approx \sum_{I=1}^{M} \sum_{J=1}^{N} F(\xi_I,\eta_J) W_I W_J \tag{4.11}$$

where M and N donate the number of quadrature points in the ξ and η directions, (ξ_I, η_J) denote the Gauss points, and W_I and W_J denote the corresponding Gauss weights. In most cases, the interpolation functions are of the same degree in both ξ and η directions, and therefore we have M = N. For quadratic rectangular element, the maximum polynomial degree is 4, the order of intergration is 3*3, and the location of integration Gauss points is $(\xi_I, \eta_J)=(\lambda, -\lambda), (-\lambda, 0), (-\lambda,\lambda), (0,-\lambda), (0, 0), (0, \lambda), (\lambda,-\lambda), (\lambda, 0), (\lambda, \lambda)$, where $\lambda=\sqrt{\frac{3}{5}}$.

The corresponding Gauss weights are 8/9 for ξ_I or $\eta_J=0$, and 5/9 for ξ_I or $\eta_J=\pm\sqrt{\frac{3}{5}}$ respectively.

It is found that if the coefficients of penalty terms are evaluated using a numerical integration rule of an order lower than that required to integrate them exactly, the finite element equations give acceptable solutions for the velocity field [38]. This technique of under-integrating the penalty terms is known as reduced integration.

In our study, we use 3*3 Gauss quadrature to calculate the non-penalty terms, and use the 2*2 Gauss quadrature to evaluate the penalty terms. The location of the integration Gauss points for penalty terms is $(\xi_I, \eta_J)=(-\lambda, -\lambda), (-\lambda, \lambda), (\lambda, -\lambda), (\lambda, \lambda)$, where $\lambda=\frac{1}{\sqrt{3}}$ and the integration weights are 1.

The choice of penalty parameter is largely dictated by the ratio of the magnitude of penalty terms to the viscous terms, the mesh, and the word length in the computer. Generally, a value of $10^4\mu$ - $10^{12}\mu$ gives good results [39][40]. It is found that the pressure is more sensitive to the value of the penalty parameter than velocity field. Here we choose the value of $10^8\mu$ for the penalty parameter.

If we express the finite element weak form of the governing equations in matrix form, they should have the following expression

$$\begin{bmatrix}[M^1][0]\\ [0][M^2]\end{bmatrix}\begin{Bmatrix}\{\dot{u}\}\\ \{\dot{v}\}\end{Bmatrix}+\begin{bmatrix}[S^{11}][S^{12}]\\ [S^{21}][S^{22}]\end{bmatrix}\begin{Bmatrix}\{u\}\\ \{v\}\end{Bmatrix}=\begin{Bmatrix}\{F^1\}\\ \{F^2\}\end{Bmatrix} \qquad (4.12)$$

for momentum conservation equations, and

$$[M^3]\{\dot{\phi}\}+[S^{33}]=\{F^3\} \tag{4.13}$$

for particle flux balance equation.

where $[M^1] = [M^2] = \mathrm{Re}(1+\varepsilon\phi)\int \varphi_i \varphi_j dx_1 dx_2$

$$[M^3] = \int \varphi_i \varphi_j dx_1 dx_2$$

$$[S^{11}] = [S^{22}] = \int (\frac{\partial \varphi_i}{\partial x_1}\frac{\partial \varphi_j}{\partial x_1}+\frac{\partial \varphi_i}{\partial x_2}\frac{\partial \varphi_j}{\partial x_2})dx_1 dx_2$$

$$[S^{33}] = \lambda \int \hat{D}_c(\phi)\dot{\gamma}(\frac{\partial \varphi_i}{\partial x_1}\frac{\partial \varphi_j}{\partial x_1}+\frac{\partial \varphi_i}{\partial x_2}\frac{\partial \varphi_j}{\partial x_2})dx_1 dx_2$$

$$[S^{12}] = [S^{21}] = \alpha \,\mathrm{Re}\,\mu \int \frac{\partial \varphi_i}{\partial x_1}\frac{\partial \varphi_j}{\partial x_2}dx_1 dx_2$$

$$[F^1] = -\mathrm{Re}(1+\varepsilon\phi)\int \varphi_i (u\frac{\partial u}{\partial x_1}+v\frac{\partial u}{\partial x_2})dx_1 dx_2$$

$$[F^2] = -\mathrm{Re}(1+\varepsilon\phi)\int \varphi_i (u\frac{\partial v}{\partial x_1}+v\frac{\partial v}{\partial x_2})dx_1 dx_2 - \frac{1}{Fr}\int \varphi_i \phi dx_1 dx_2$$

$$[F^3] = -\int \lambda (D_s(\phi)(\frac{\partial \varphi_i}{\partial x_1}\frac{\partial \dot{\gamma}}{\partial x_1}+\frac{\partial \varphi_i}{\partial x_2}\frac{\partial \dot{\gamma}}{\partial x_2})+\frac{2}{9Fr}\phi f(\phi))+\varphi_i(u\frac{\partial \phi}{\partial x_1}+v\frac{\partial \phi}{\partial x_2}))dx_1 dx_2$$

4.5 Time Approximations

This laminar suspension flow is a time dependent problem, we need to derive the time approximation to convert the ordinary differential equations in time to algebraic equations.

If we write the eq. (4.13) as

$$[M]\{\dot{u}\}+[S]\{u\} = \{F\} \tag{4.14}$$

and the initial condition is

$$\{u\}_0 = \{u_0\} \tag{4.15}$$

where $\{u\}_0$ means the value of u at time t=0, and $\{u_0\}$ is a given value.

The most commonly used method for solving (4.15) is the α family approximation. A weighted average of the time derivative of the dependent variable is approximated at two consecutive time steps by linear interpolation of the values of the variable at these two steps.

$$(1-\alpha)\{\dot{u}\}_s + \alpha\{\dot{u}\}_{s+1} = \frac{\{u\}_{s+1} - \{u\}_s}{\Delta t_{s+1}} \tag{4.16}$$

For different value of α, there are different numerical integration schemes. In our study, we choose α=1, i.e. the backward difference scheme, then the eq. (4.15) becomes

$$([M] + \Delta t_{s+1}[S])\{u\}_{s+1} = [M] + \Delta t_{s+1}\{F\}_{s+1} \tag{4.17}$$

This equation can be solved with initial conditions and the value of {F} at time step s+1, which could be obtained using iteration.

In our study, the time step is set to be 0.01.

4.6 Explanations for the Computer Program

The FEM analysis is performed with a C++ program, the procedure for the analysis entails the follow steps

1. It is assumed that all the dependent variables are known at time step t_n.
2. The concentration ϕ is calculated at time t_{n+1} from Eq. (4.4), using values of all the variables and of the diffusion coefficients at time t_n.

3. the velocity components u and v of the suspension flow are calculated from Eq.(4.3) using the updated values of ϕ obtained in step 2.
4. The effective shear rate $\dot{\gamma}$ is calculated from Eq. (3.8) using all of the updated values.
5. The process is then begin anew starting from step 1.

Our finite element program consists of three basic parts,

1. Preprocessor,
2. Processor,
3. Postprocessor.

In the preprocessor, the input data of the problem are read and generated. This includes the geometry, the data of the problem, finite element mesh information (e. g. element type, number of elements, element length, coordinates of the nodes, and connectivity matrix).

In the processor part, all steps in the finite element method, except for postprocessor, are performed. These include the follow

1. Generation of the element matrices using numerical integration.
2. Assembly of the element equations.
3. Imposition of the boundary conditions.
4. Solution of the algebraic equations for the nodal values of the primary variables.

In the postprocessor part, the effective shear rate is computed by interpolation at Gauss points. And the output data are written in the file.

The major functions in the program are

CALNODE: calculate the node connection matrix for each element.

CAL_COORX: calculate the x coordinates of the nodes in each element.

CAL_COORY: calculate the y coordinates of the nodes in each element.

DEFINE_0: class function of CLASS ONE_ELEMENT, initialize the coefficient.

DEFINE_MATRIX0: class function of CLASS ONE_ELEMENT, initialize the element matrix.

CAL_FAI: calculate the interpolation functions.

CAL_KC1: calculate the first order derivation of interpolation function respect to ξ direction.

CAL_KC2: calculate the second order derivation of interpolation function respect to ξ direction.

CAL_AT1: calculate the first order derivation of interpolation function respect to η direction.

CAL_AT2: calculate the second order derivation of interpolation function respect to η direction.

CAT_KCAT: calculate the derivation of interpolation function respect to ξ and η directions.

CAL_J: calculate J matrix, the Jacobian matrix of coordinate transformation.

CAL_INVJ: calculate the invert matrix of matrix J.

CAL_J1: calculate J1 matrix, the second order Jacobian matrix.

CAL_INVJ1: calculate the invert matrix of J1.

CAL_F: class function of CLASS ONE_ELEMENT, calculate the matrix on the right side of the algebraic equations.

CAL_AE1: class function of CLASS ONE_ELEMENT, calculate the matrix on the left side of the algebraic equations.

CAL_GLOBAL1: class function of CLASS ONE_ELEMENT, calculate the assembly matrix.

CAL_BOUNDARY1: impose the boundary conditions on the assembly matrix.

CAL_SOLVE1: solve the linear algebraic equations.

CHAPTER 5

RESULTS AND DISCUSSION

In this chapter we will discuss numerical results for the viscous suspension flow in a two-dimensional channel. First some profiles of the velocity and concentration are presented. Next the effects of each parameter on the flow are discussed. Finally the numerical results are compared with existing experimental results.

5.1 Finite Element Mesh and Node Numbering

When a uniform flow enters a channel, the length of the channel required for the flow to become fully developed (e. g. the entrance length Le) is a function of Reynolds number [41],

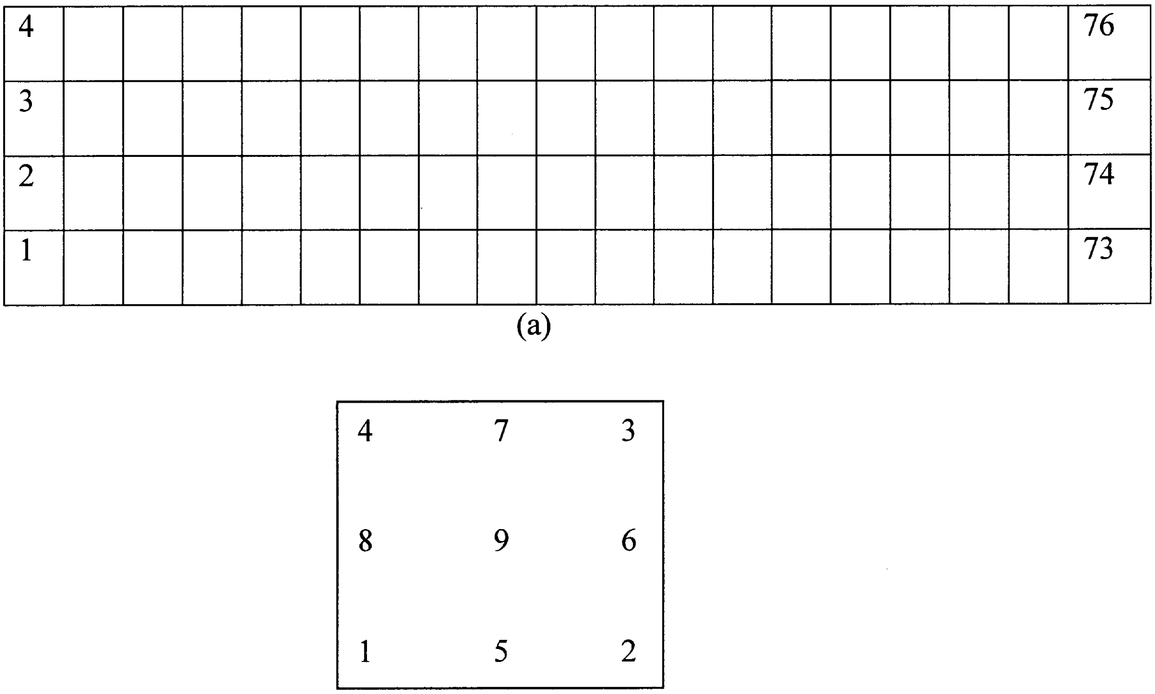
$$\frac{Le}{H} = \frac{0.79}{0.04\,\mathrm{Re}_f + 1} + 0.053\,\mathrm{Re}_f \tag{5.1}$$

where Re_f is based on the flow velocity and the channel height h or $\mathrm{Re}=\rho_f U h/\mu_f$. However in the presence of suspended particles, the coupled velocity and concentration profiles develop much slower. Nott and Brady [24] suggested that the steady-state concentration profile prevailed only at a fairly long development length ratio, L/h, which is roughly estimated to be on the order of

$$\frac{L}{h} = \frac{1}{2}\left(\frac{h}{d}\right)^2 \tag{5.2}$$

for concentrated suspension ($\phi > 0.3$). In Equation (5.2) L is the particle concentration development length, h is the height of the flow channel, and d is the diameter of particle. Recent experiments by Hampton [20] indicate that this development length ratio should be from one-third to one-half of this prediction. In our study, h/d ranges from 8.19 to 33.4 and therefore from Eq. (5.2) L/h ranges from 33.5 to 558. Taking one-third of the predicted value, we have $11<L/h<186$.

Depending on the L/h ratio, we have divided the length of flow channel into 19 to 79 elements horizontally, and the height of the flow channel is divided into 4 elements vertically. Thus we have 76 to 316 elements, and 351 to 1431 nodes. With 3 degrees of freedom for each node, we have a total of 1053 to 4293 degrees of freedom.



(a)

(b)

Figure 5.1(a)(b) Finite element mesh and element node numbering order.

Figure 5.1 shows the finite element mesh and element node numbering order. The node of each element is numbered as shown in fig. 5.1(b). Node 1-4 are the corner nodes, node 5-8 are the side nodes, and node 9 is the center node. Figure 5.1(a) shows the order of element numbering. The nodes are numbered along the column so that the half band width is minimized.

The boundary and initial conditions are as given in chapter 3. Basically we consider a uniform suspension flow enters the channel with uniform velocity.

Figure 5.2 shows the result profiles of velocity and concentration for concentrated suspensions for ϕ_0 =0.3, Re=37 at the exit of the channel. From Figure 5.2 (a) and (b) we can see that for concentrated suspension ($\phi \geq 0.3$), under the action of gravity, a region of clear fluid appears at the top of the flow channel. The range of this region will change with different flow parameters, as we will discuss it later. We can also see that the axial velocity profiles are blunt compare with the axial velocity of pure Newtonian fluid, which is a fully-developed Poiseuille flow. The axial velocity of the suspension is sheared toward the upper wall. The most important feature is that the particle concentration does not decrease monotonically from the bottom to the top of the channel. Instead, a high concentration region is formed above the channel centerline. The mechanism for the formation of this higher concentration region is that when the average inlet particle concentration is high, the diffusion due to the shear-rate gradient is strong and creates a large upward flux of particles. On the other hand, when this part of shear-induced diffusion is not strong enough, i.e. the initial particle concentration is relatively low, the increase in particle concentration above the centerline will not be that pronounced. We will also discuss the effects of inlet concentration in the following sections.

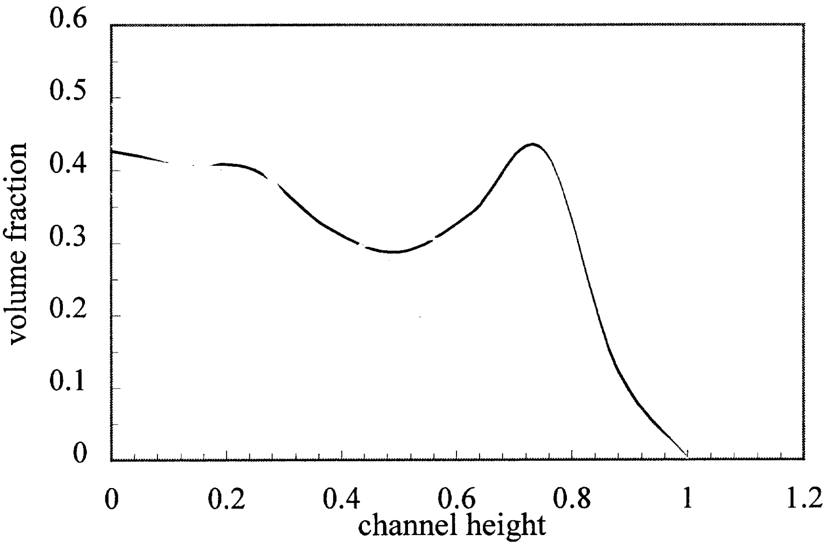


Figure 5.2 (a) Particle concentration profile for inlet concentration ϕ_0 =0.3, Re=46, d/H=0.003.

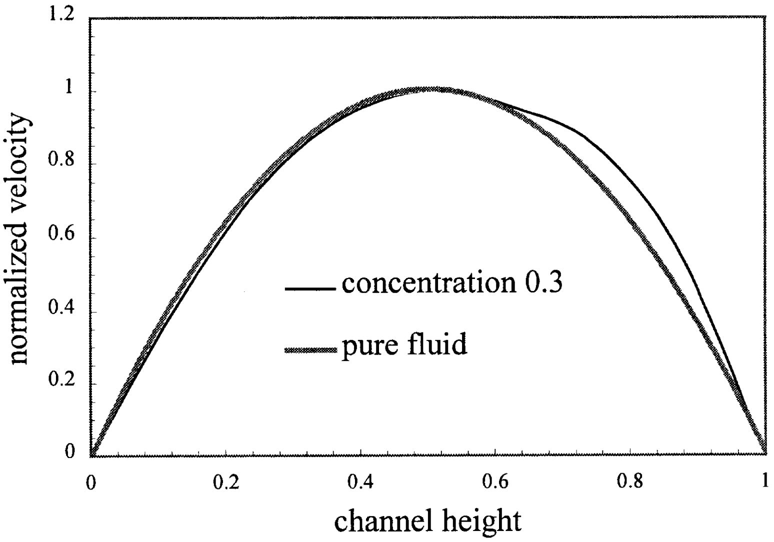


Figure 5.2 (b) X-axis velocity profile for inlet concentration ϕ_0 =0.3, Re=46, d/H=0.003.

5.2 Ranges of Parameters in The Numerical Calculations

In this study the dimensionless parameters are inlet concentration ϕ_o, ratio of particle diameter to channel height d/h, Reynolds number Re (= $\rho u_o h/\mu_f$), Froude number Fr ($u_o\mu_f/[\{\rho_p-\rho_f\}h^2g]$), and electrostatic charge number $Q=\frac{(\rho_p-\rho_f)\rho_f\rho_p h^4}{\mu^2{}_f\varepsilon_0}\left(\frac{q}{m_p}\right)^2$. The variation of these parameters employed in this study is listed in Table 5.1. The reasons for chosen these values are explained as follows.

Existing experimental studies [17] have shown that the inlet concentration of about 0.3 would give a concentration peak above the channel centerline. We will study the effect of the inlet concentration on this peak development and select ϕ_o = 0.20, 0.25, 0.30 and 0.35.

In previous studies [27,31] there were some simple reports on the effect of d/h ratio on the flow but so far no numerical investigations were reported. The particles used in their study have the diameter of 762 μm and 3175 μm. The channel has the height of 25.4mm. Thus the d/h ratios selected in this study are 0.03, 0.06, 0.09 and 0.12.

For a given type of particle flowing in a fluid in a channel we have physical properties such as h, g, ρ_p, ρ_f, and μ_f defined. The other physical variable, velocity, can be selected and both Re and Fr are determined. Consider particles with ρ_p = 1003 kg/m^3 flow in an oil at 20 °C and 1 atmospheric pressure with ρ_f =875 kg/m^3 , μ_f=3.86P in a channel with h=0.0254 m. With u_o = 0.8 (m/s), 0.64, 0.48, 0.32, 0.2, 0.1, we have Re = 46, 37, 27, 18, 11.5, 5.75 and Fr=0.038, 0.30, 0.23, 0.15, 0.095, 0.0475 respectively. These Re and Fr are used in the numerical simulations.

When particles are mixed with fluid or flowing in a conduit they may collide with walls and electrostatic charges are generated. Particles will accumulate charges and produce repulsive forces among them. Though the amount of charges is very small, the effect on the particle concentration may be significant. To study this effect we include the electrostatic charge parameter Q which is defined in Eq. (5.9). Q ranges from 0.0001 to 0.003.

Table 5.1 shows the parameter values we used in our numerical study, and we will discuss the effect of each parameter in the following sections.

Table 5.1 Values for each parameter in the numerical study

Inlet concentration ϕ_0	0.2	0.25	0.3	0.35
Ratio of particle diameter to channel height (d/h)	0.12, 0.09, 0.06, 0.03	0.03	0.12, 0.09, 0.06, 0.03	0.03
Electric Field Q	0, 0.001, 0.0005, 0.0001	0	0, 0.003 0.001 0.0005, 0.0001	0, 0.001 0.0005, 0.0001
Modified Froude number (Fr)	0.38, 0.3, 0.23, 0.15, 0.095, 0.0475	0.38, 0.3, 0.23, 0.15, 0.095, 0.0475	0.38, 0.3, 0.23, 0.15, 0.095, 0.0475	0.38, 0.3 0.23, 0.15, 0.095, 0.0475
Reynolds number (Re)	46, 37, 27, 18, 11.5, 5.75	46, 37, 27, 18, 11.5, 5.75	46, 37, 27, 18, 11.5, 5.75	46, 37, 27, 18, 11.5, 5.75

5.3 Developing Flow in the Entrance Region

Unlike other investigations in which parabolic inlet velocity profiles were employed we have considered uniform inlet velocity profile. In this analysis both the hydrodynamic and particle concentration are developing in the channel.

It is well known that the hydrodynamic entrance length is relatively short in a pure fluid flow. As a mater of fact the results showed that the velocity profile becomes parabolic within two to three channel lengths and after that the velocity profile makes small changes to accommodate the particle concentration change. Theoretically the velocity

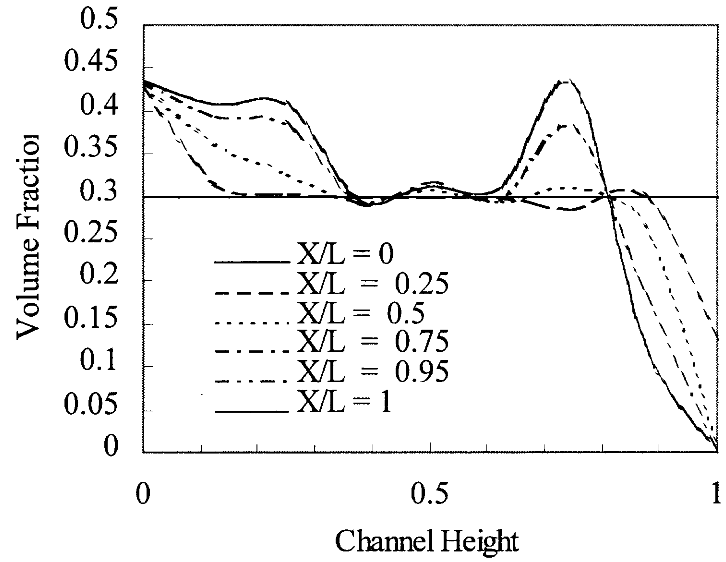


Figure 5.3(a) Development of particle concentration with ϕ_0=0.3, Re=46, Fr=0.38, d/H=0.03.

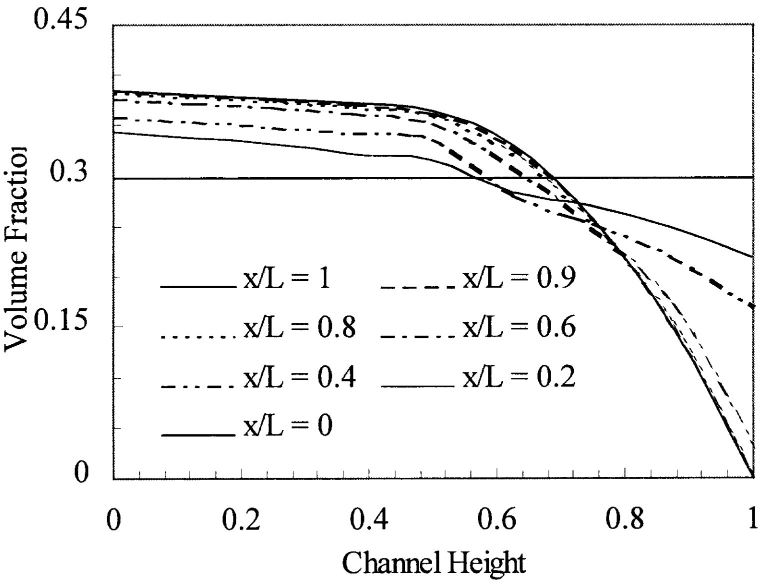


Figure 5.3(b) Development of particle concentration with ϕ_0=0.3, Re=46, Fr=0.38, d/H=0.12.

profile changes as the concentration profile changes and become steady when the concentration profile is fully-developed. Figure 5.3(a) depicts particle concentration development in the channel for ϕ_o=0.3, Re=46 and d/h=0.03 with the channel length of 160. Near the inlet region the fluid velocity changes from uniform to parabolic profile while particles fall down toward the bottom due to gravity. As the flow proceeds viscous shear effect starts to push particles toward the centerline and two maximum concentrations appear in the flow. Eventually the concentration in the centerline region also shows slightly higher than the average concentration. Figure 5.3(b) shows the particle concentration development in the flow channel with ϕ_o=0.3, Re=46 and d/h=0.12 with the channel length of 20. The particle concentration distribution has big difference between Figure 5.3(a) and Figure 5.3(b), and this is due do particle/channel height ratio, we will discuss it in section 5.7. With higher d/H ratio, the developing length is much shorter than that with smaller ratio. The particle concentration decrease monotonically in the whole flow channel.

5.4 Fully-Developed Concentration Profile

Before we discuss the effects of various parameters on the particle concentration, it is of interest to investigate if the particle concentration has reached its fully-developed profile at the end of the channel in this study. To obtain a fully-developed profile we delete the x-derivative terms in the governing equations. Figure 5.4(a) compares the concentration profile obtained from developing flow analysis with L=160 and that obtained from the fully-developed flow analysis. Both flows are at ϕo=0.30, Re=27 and d/h=0.03. Figure 5.4(b) is the comparison of particle concentration of fully developed flow and developing flow at ϕo=0.30, Re=46 and d/h=0.03. It is seen that the two analyses give good agreement. It is possible to conclude that for Re $\leq$ 46 and $\phi o \leq 0.3$ the present analysis should give a fully-developed flow at the end of the channel. For Re=46 and $\phi o=0.35$ the flow may not have reached its fully-developed condition.

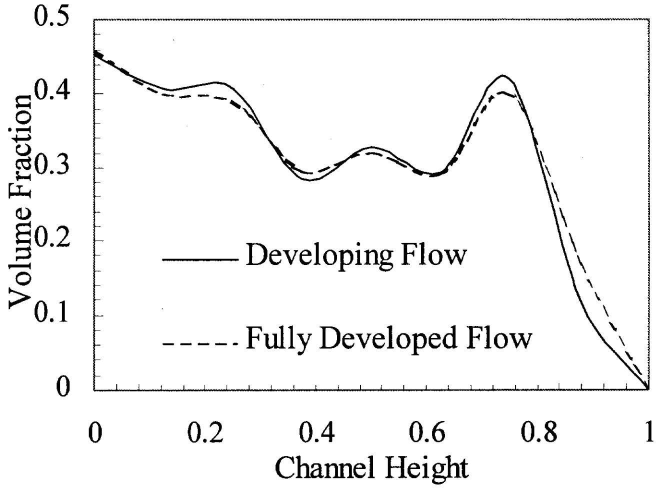


Figure 5.4 (a) Particle concentration distribution for developing flow at L=160 and fully developed flow with $\phi_o=0.30$, Re=37, d/H=0.03.

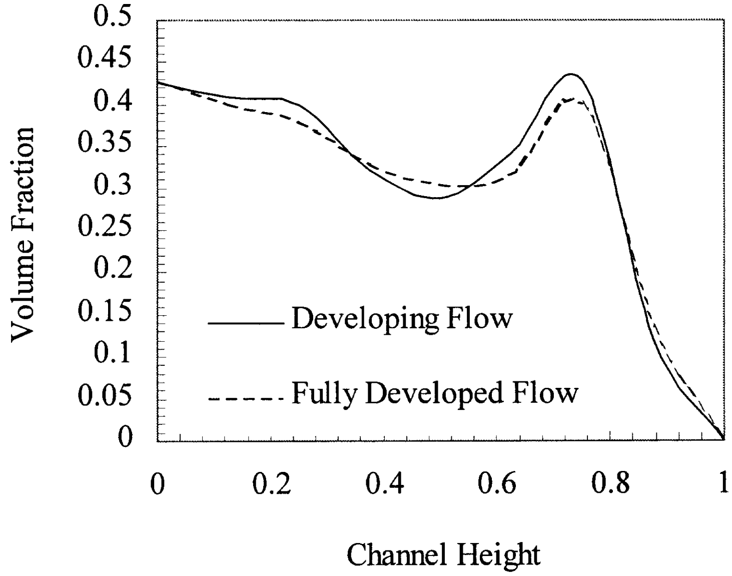


Figure 5.4 (b) Particle concentration distribution for developing flow at L=160 and fully developed flow with $\phi_o=0.30$, Re=46, d/H=0.03.

5.5 Effects of Initial Concentration on Velocity and Concentration Profiles

When a fluid suspended with uniformly distributed particles flows into a parallel-plate channel with uniform velocity, both velocity and particle concentration will developed into fully developed profiles. The length of channel employed in this analysis ranges from 20 to 160 depending on the ratio of the particle diameter to the channel width. Figure 5.5 depicts profiles of normalized velocity U/U_c at the exit plane of a channel with $L/h= 160$ at $Re=46$, $Fr=0.38$ and $d/h= 0.03$ for $\phi_o=0.35$, 0.30, 0.25 and 0.20 respectively. The normalized velocity U/Uc is the ratio of the local velocity U to the centerline velocity U_c.

As is seen from the figure the velocity profile at $\phi_o=0.20$ is practically equal to that of a Poiseuille flow (or parabolic profile). As ϕ_o is increased from 0.20 to 0.35 the profile for the lower portion of the channel shows very little changes from the parabolic profile but the top portion of the channel shows higher velocity than that of the parabolic profile. For example at $X_2=0.75$, U/U_c is 0.75 and 0.83 for $\phi_o= 0.20$ and 0.35 respectively and the increase in velocity is 11%. The centerline velocity U_c changes from 1.50 at $\phi_o=0.20$ to 1.42 at $\phi_o = 0.35$. The deduction in the centerline velocity is required in order to satisfy the continuity equation. Similar increase in velocity in the top portion of channel is also found in flows at other Reynolds numbers.

Figures 5.6(a-d) present normalized concentration ϕ/ϕ_o for $\phi_o =0.20$, 0.25, 0.30 and 0.35 at $Re = 46$, 37, 27 and 18 ($Fr = 0.38$, 0.30, 0.23 and 0.15) respectively. At any given Reynolds number the maximum ϕ/ϕ_o in the upper region of the channel increases as ϕ_o increases from 0.20 to 0.30 and then decreases as V is increased to 0.35. This is also the case for the small peak in the lower region of the channel. The centerline concentrations are about 5 to 10 % higher than the inlet concentrations.

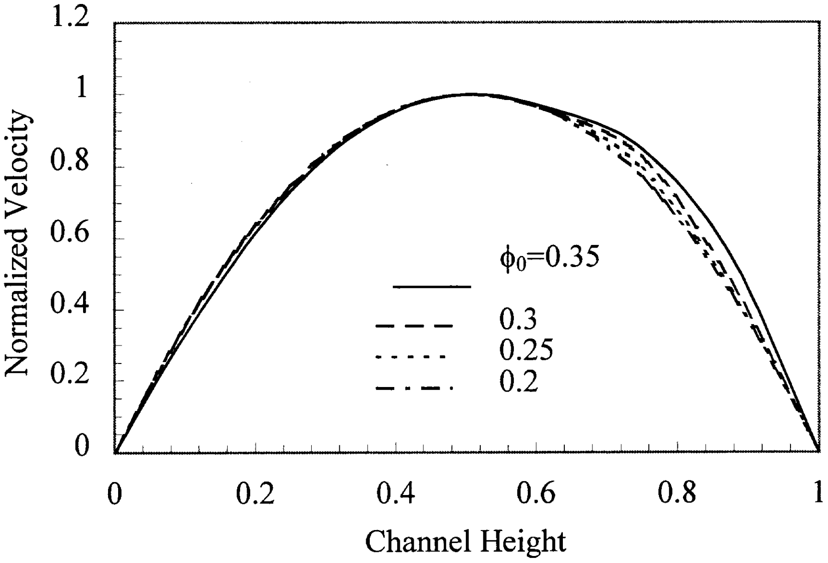


Figure 5.5 Normalized velocity at channel exit with Re = 46, Fr = 0.38, d/H=0.03.

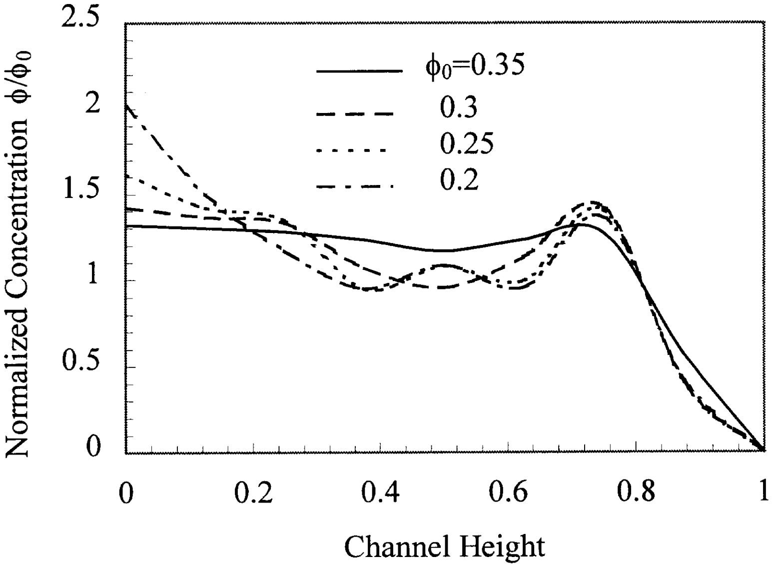


Figure 5.6 (a) Particle concentration distribution Re=46, Fr=0.38, d/H=0.03.

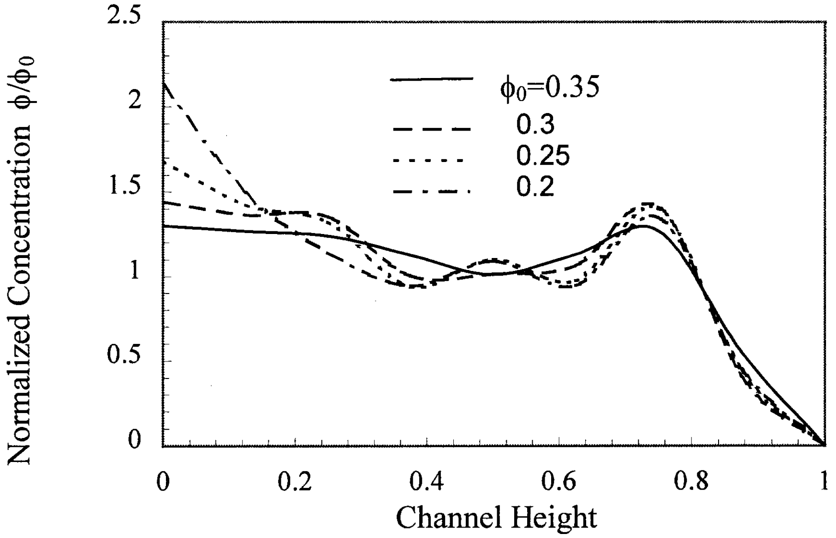


Figure 5.6 (b) Particle concentration distribution at Re=37, Fr=0.3, d/H=0.03.

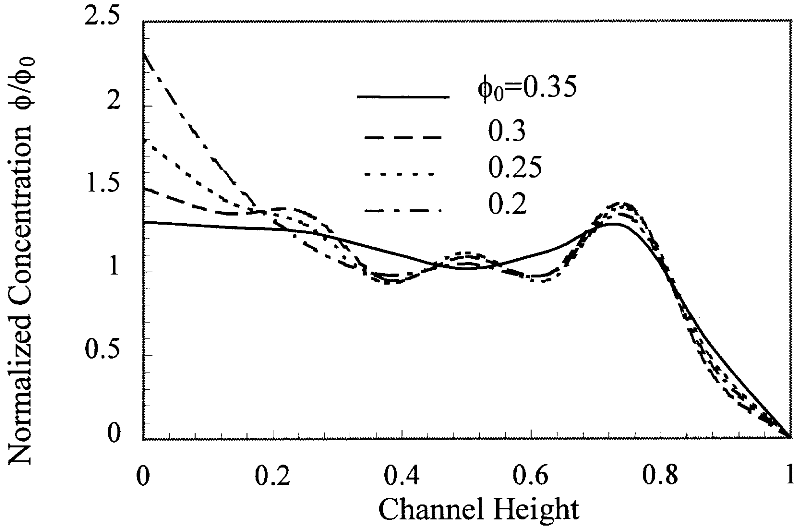


Figure 5.6 (c) Particle concentration distribution at Re=27, Fr=0.23, d/H=0.03.

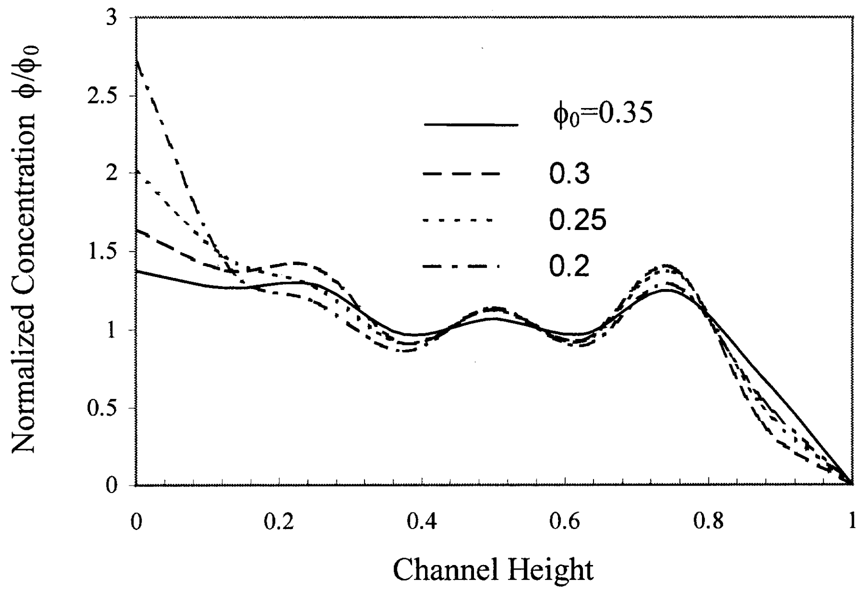


Figure 5.6 (d) Particle concentration distribution at Re=18, Fr=0.15, d/H=0.03.

5.6 Effects of Reynolds Number and Modified Froude Numbers

As explained in section 5.2 values of velocity were changed from 0.8 to 0.1 m/s which resulted in changes of Reynolds number (and Froude numbers as well) from 46 (0.038) to 5.75 (0.0475). Therefore higher Reynolds number implies higher velocity in this analysis.

Figure 5.7(b) depicts velocity distribution at the exit plane of the channel at ϕ_o=0.35 under various Reynolds numbers. The deviation from parabolic profile at upper portion of the channel is greatest for Re=46. This phenomenon of shear induced diffusion is also found in other inlet concentrations.

Figure 5.7(1a) presents particle concentration at the channel exit for ϕ_o=0.35 under various Reynolds numbers at d/h=0.03. At this high inlet particle concentration the concentration profiles for $46 \leq Re \leq 27$ are relative flat and they may not have been fully developed yet. Figure 5.7(2a) presents particle concentration at the channel exit for $\phi_o=0.30$ under various Reynolds numbers at d/h=0.03. The profiles are quite similar and except that for $\phi_o=0.35$ all curves exhibit three maximums in the flow. When the viscous shear effect on particles is neglected we obtained a monotonically decreasing concentration from the top wall toward the bottom wall as shown in the figure with a heavy solid line. This results is the same as that of [26].

Figures 5.7(3a,4a) present the results for particle concentration at the channel exit for $\phi_o=0.25$ and $\phi_o=0.20$ under various Reynolds numbers at d/h=0.03. The concentration distributions are very close together indicating that the effects of Reynolds number (or more precisely the effects of velocity) on the concentration profile is small when $0.25 \leq \phi_o$.

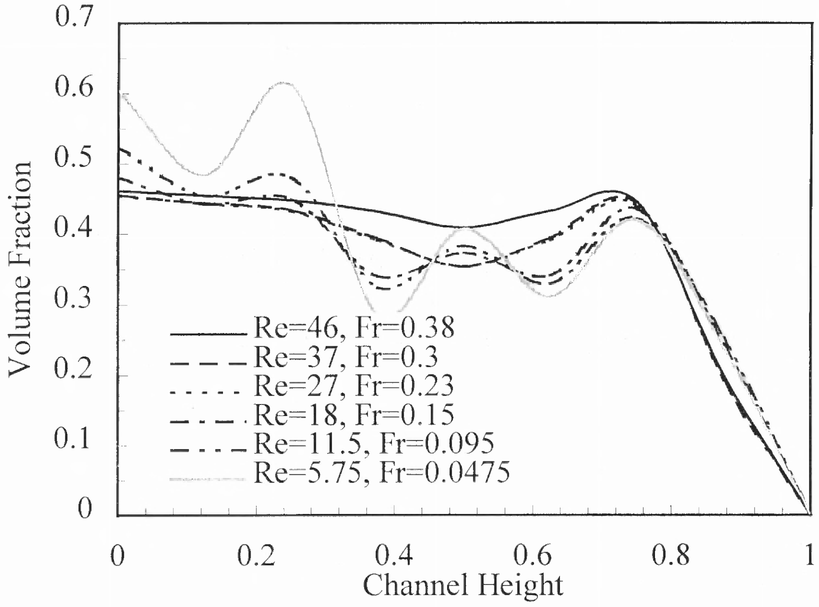


Figure 5.7(1a) Particle concentration at inlet particle concentration $\phi_0 = 0.35$,d/H=0.03.

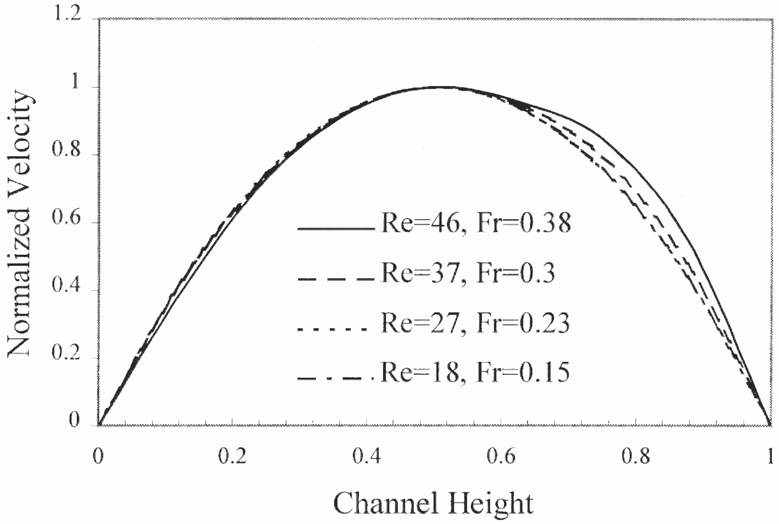


Figure 5.7(1b) Axial velocity at inlet particle concentration $\phi_0 = 0.35$, d/H=0.03.

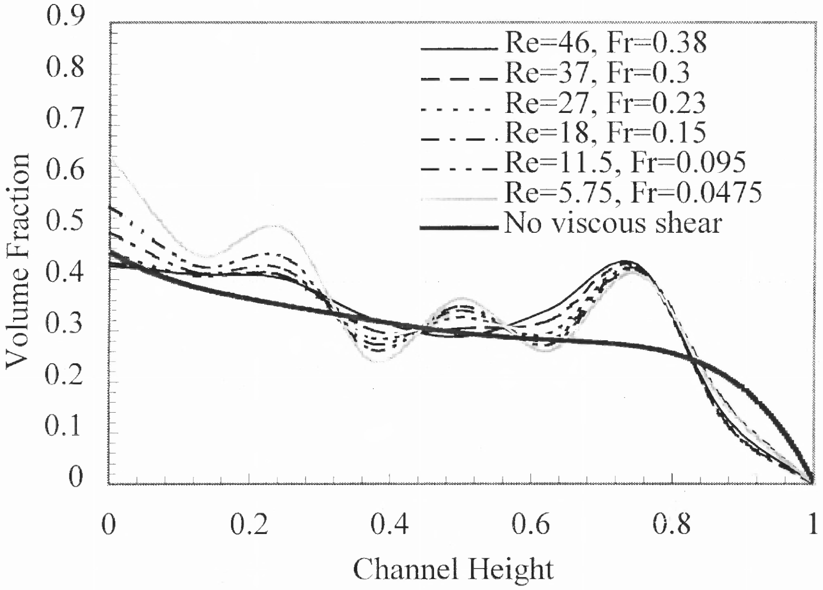


Figure 5.7(2a) Particle concentration at inlet particle concentration $\phi_0 = 0.3$, d/H=0.03.

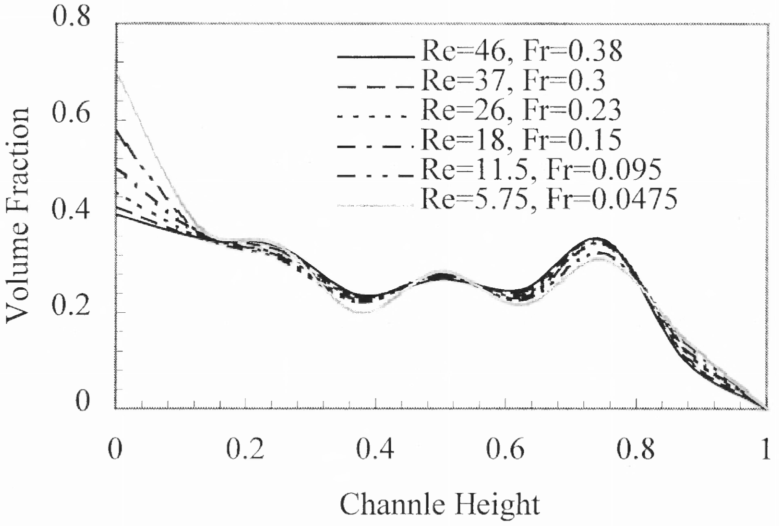


Figure 5.7(3a) Particle concentration at inlet particle concentration $\phi_0 = 0.25$, d/H=0.03.

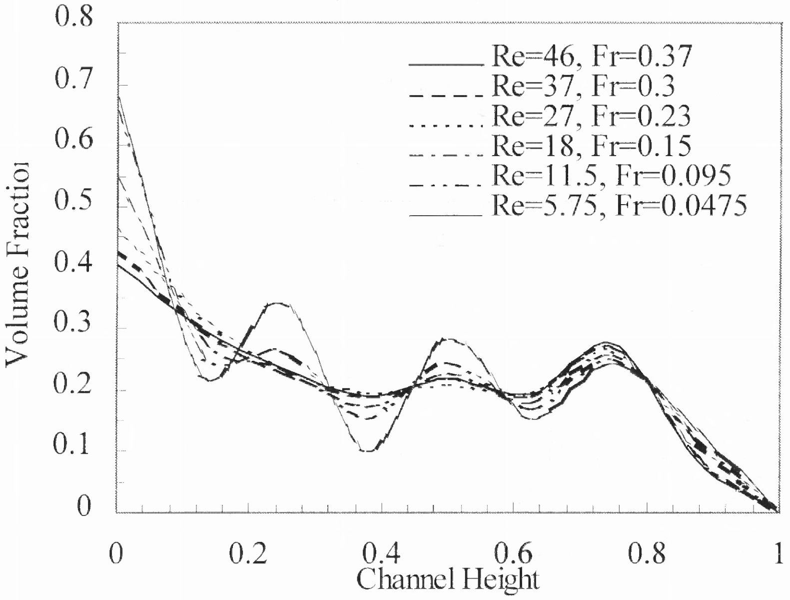


Figure 5.7(4a) Particle concentration at inlet particle concentration $\phi_0 = 0.2$, d/H=0.03.

5.7 Effects of Ratio of Particle Diameter to the Channel Height

There are some experimental studies on the effects of ratio of particle diameters to the channel height. Miskin et al. [26] [27] did some researches on this ratio in an inclined rectangular channel. Hampton et al. [20] also conducted some experiments on this ratio on the demixing of neutrally buoyant suspension of spheres in slow, pressure driven flows in circular couduits. In our numerical simulation, we tested several different value of this ratio: d/h=0.12, 0.9, 0.06, 0.03, where d is the diameter of particle, and h is the height of the flow channel.

Figures 5.8(1a-2a) show the particle concentration under different inlet concentration. We can see that at smaller d/h, the particle concentration above centerline is very high, or even cusp on the concentration distribution curve, the highest particle concentration position is above the centerline. But at bigger d/h ratio, the particle concentration distribution curve is smoother, and the particle concentration above centerline is only a little bit higher than that at lower location, the highest particle concentration position is not above the centerline. This means that when d/H ratio decrease, the effect of viscous shear lift force on the particles increase and produce a maximum concentration above the centerline of the channel. Figure 5.8(1a) shows the particle concentration distribution under the inlet concentration of 0.3, we can see at the d/H ratio of 0.12, the particle concentration almost decrease monotonically. Figure 5.8(2a) shows the particle concentration under the inlet particle concentration of 0.2, we can't see the concentration decrease monotonically, this means that the d/h ratio has more effect on high inlet particle concentration. Figure 5.8(2a, 3a) shows the particle concentration under different Re number. We can see under higher Re number, this ratio has bigger effects on the particle concentration. With the d/h ratio of 0.12, the particle concentration decrease almost monotonically, but with the d/h ratio of 0.03, we can see the concentrated region above the centerline clearly. But for lower Re number, this ratio doesn't affect the particle concentration a lot. Figure 5.8(1b, 3b) shows the axial velocity profiles under inlet concentration of 0.3, Re=46 and $\phi_0=0.2$, Re=18 respectively. We can see that this d/h ratio does not change the velocity profiles very much. At smaller d/h ratio, the velocity profile is more asymmetric, and at bigger d/h ratio, the axial velocity profile is almost the parabolic curve.

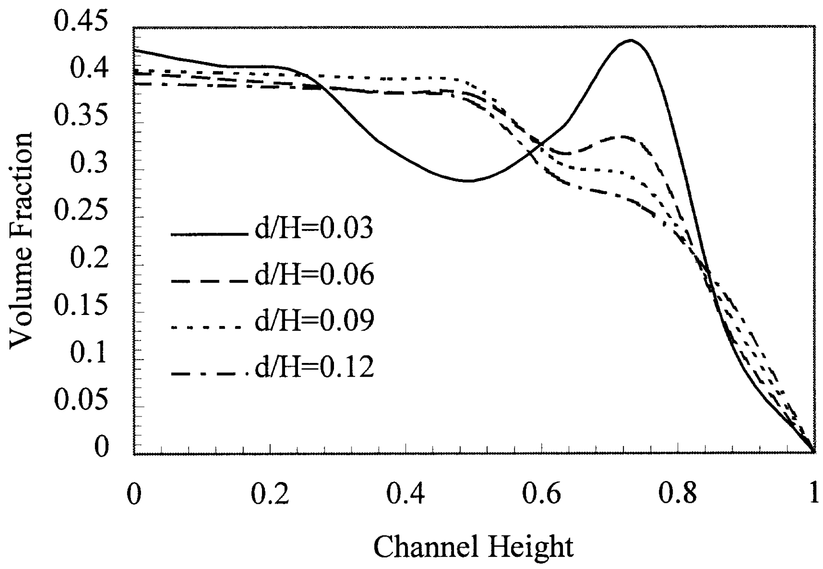


Figure 5.8 (1a) Effect of d/h on ϕ distribution at inlet concentration of 0.3, Re=46.

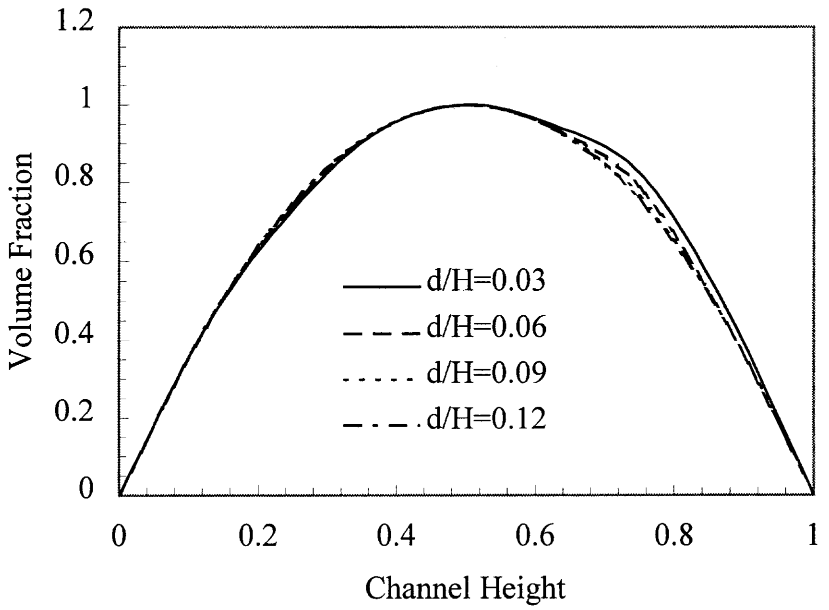


Figure 5.8 (1b) Effect of d/h on axial velocity distribution at ϕ_0=0.3, Re=46.

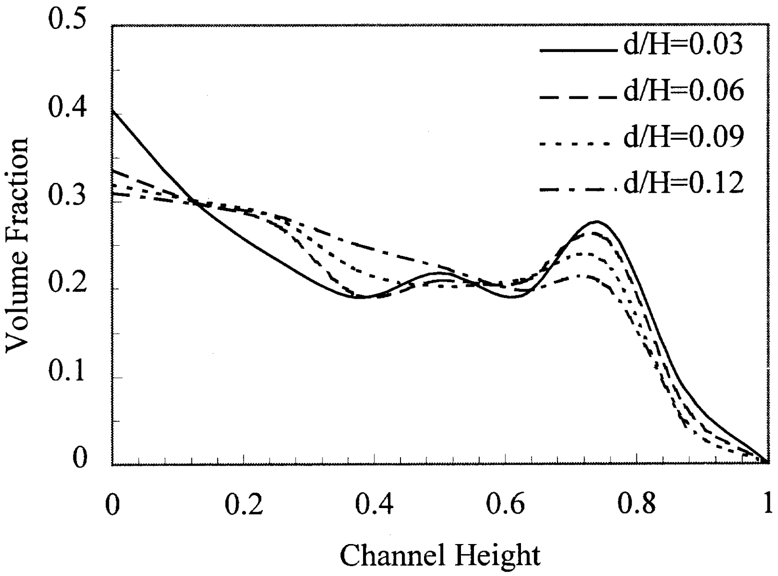


Figure 5.8 (2a) Effect of d/h on ϕ distribution at ϕ_0 =0.2, Re=46.

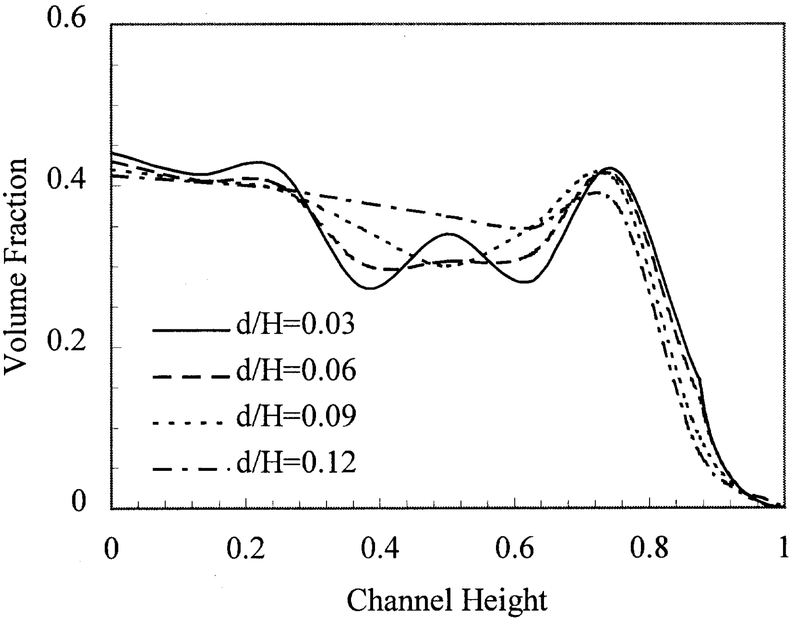


Figure 5.8 (3a) Effect of d/h on ϕ distribution at Re=18, ϕ_0=0.2 .

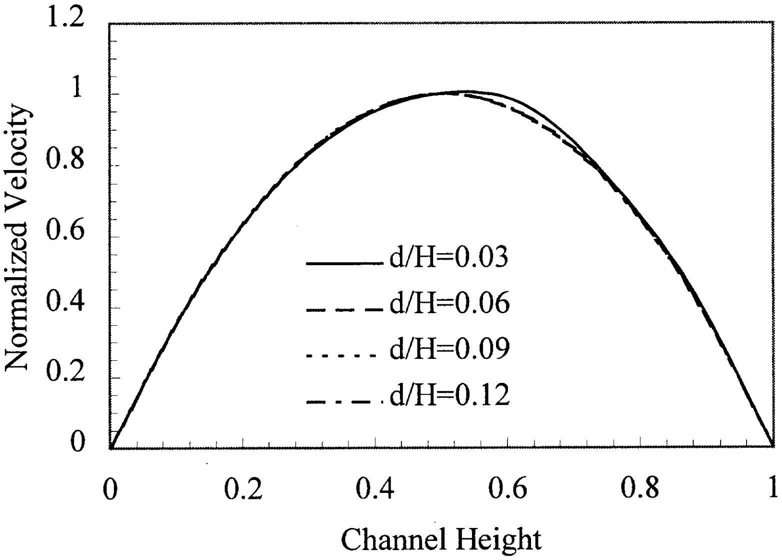


Figure 5.8 (3b) Axial velocity under different d/h value at Re=18, ϕ_0=0.2.

5.8 The Effect of Electric Field

The particle distribution of the viscous suspensions can be influenced by electrostatic charges on the particles. The electrostatic charges on the moving particles are generated by collision among the solid particles and also by collision between the solid particles and the walls prior to their entrance into the channel. Unlike particles suspended in air the amount of charges accumulated on the particles are relatively small for suspensions in a liquid and thus the electrostatic parameter is very small. In our study, we will discuss the effect of this electrostatic parameter on the viscous suspension flow.

The governing equation in Chapter 3 is modified by adding the electrostatic force and electro-induced migration to momentum and particle conservation equations respectively.

The electrostatic force is

$$F_E = \phi \rho_p \frac{q}{m_p} \frac{dw}{dx_2} \tag{5.3}$$

the electrostatic-induced migration is

$$N_E = \frac{2}{9} \frac{a^2 (\rho_p - \rho_f)}{\mu_f} \phi f(\phi) \frac{q}{m_p} \phi \rho_p \frac{\partial w}{\partial x_2} \tag{5.4}$$

where the electrostatic field potential w is governed by

$$\frac{\partial^2 w}{\partial x_1^2} + \frac{\partial^2 w}{\partial x_2^2} = -\frac{\rho_p}{\varepsilon_0} \frac{q}{m_p} \tag{5.5}$$

where

m_p is the mass for each particle

q is the electric charge for each particle.

ε_0 is permittivity of free space.

The governing equations become

$$\rho(\phi)\frac{DU}{dt} = \nabla \bullet \sigma + (\rho(\phi) - \rho_f)g + \phi\rho_p \frac{q}{m_p}\frac{dw}{dx_2} \tag{5.6}$$

$$\frac{D\phi}{Dt} = \nabla \cdot (-a^2\dot{\gamma}(K_c\phi + K_\mu\phi^2\frac{1}{\mu}\frac{d\mu}{d\phi})\nabla\phi - a^2K_c\phi^2\nabla\dot{\gamma}$$

$$+\frac{2}{9}\frac{a^2(\rho_p - \rho_f)}{\mu_f}\phi f(\phi)g$$

$$+\frac{2}{9}\frac{a^2(\rho_p - \rho_f)}{\mu_f} f(\phi)\frac{q}{m_p}\phi\rho_p\frac{\partial w}{\partial x_2}) \tag{5.7}$$

$$\frac{\partial^2 w}{\partial x_1^2} + \frac{\partial^2 w}{\partial x_2^2} = -\frac{\rho_p}{\varepsilon_0}\frac{q}{m_p} \tag{5.5}$$

The dimensionless forms for these governing equations become

$$\mathrm{Re}(1+\varepsilon\phi)(\frac{\partial U_i}{\partial t} + U_j\frac{\partial U_i}{\partial X_j}) = \frac{\partial}{\partial X_j}[-p\delta_{ij} + 2\mu(\phi)d_{ij}] - \frac{1}{Fr}\phi\delta_{i2} - Q\frac{1}{\mathrm{Re}}\frac{\rho_p}{(\rho_p - \rho_f)}\frac{\partial W}{\partial X_2}\delta_{i2}$$

i=1,2, j=1,2 (5.8)

$$\frac{\partial\phi}{\partial t} + U_j\frac{\partial\phi}{\partial X_j} = \lambda\frac{\partial}{\partial X_j}[\hat{D}_C(\phi)\dot{\gamma}\frac{\partial\phi}{\partial X_j} + \hat{D}_S(\phi)\frac{\partial\dot{\gamma}}{\partial X_j}$$
$$+\frac{2}{9\,Fr}\phi f(\phi)\delta_{j2} + \frac{2}{9}Q\frac{1}{\mathrm{Re}}\phi f(\phi)\frac{\partial W}{\partial X_2}\delta_{j2}]$$

j = 1,2 (5.9)

$$\frac{\partial^2 W}{\partial X_1^2} + \frac{\partial^2 W}{\partial X_2^2} = -\phi \tag{5.10}$$

and

$$Q = \frac{(\rho_p - \rho_f)\rho_f \rho_p h^4}{\mu^2{}_f \varepsilon_0}\left(\frac{q}{m_p}\right)^2 \tag{5.11}$$

W is the dimensionless form of electrostatic field potential.

$$W = \frac{w\varepsilon_0 m_p}{\rho_p q h^2} \tag{5.12}$$

All the other parameters are the same as those in chapter 3.

The weak form of the governing equations can be written as

$$\int_\Omega \omega(1+\varepsilon\phi)\left(\frac{\partial U_i}{\partial t} + U_j\frac{\partial U_i}{\partial X_j}\right)d\Omega + \frac{1}{\mathrm{Re}}\int_\Omega \mu\frac{\partial \omega}{\partial X_j}\left(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i}\right)d\Omega - \int_\Omega \mu\frac{\partial \omega}{\partial X_i}\frac{\partial U_j}{\partial X_j}d\Omega$$

$$= -\frac{1}{\mathrm{Re}\ Fr}\int_\Omega \omega\phi\delta_{i2}\,d\Omega - \int \omega Q\frac{1}{\mathrm{Re}^2}\frac{\rho_p}{\rho_p - \rho_f}\frac{\partial W}{\partial X_2}\delta_{i2}\,d\Omega \tag{5.13}$$

$$\int_\Omega \varpi\left(\frac{\partial \phi}{\partial t} + U_j\frac{\partial \phi}{\partial X_j}\right)d\Omega + \lambda\int_\Omega \frac{\partial \varpi}{\partial X_j}\left[\hat{D}_C(\phi)\dot{\gamma}\frac{\partial \phi}{\partial X_j} + \hat{D}_s(\phi)\frac{\partial \dot{\gamma}}{\partial X_j}\right.$$

$$\left. + \frac{2}{9\,Fr}\phi f(\phi)\delta_{j2} + \frac{2}{9}Q\,\frac{1}{\mathrm{Re}}\phi f(\phi)\frac{\partial W}{\partial X_2}\delta_{j2}\right]d\Omega = 0 \tag{5.14}$$

$$\int\left(\frac{\partial \Psi}{\partial X_1}\frac{\partial W}{\partial X_1} + \frac{\partial \Psi}{\partial X_2}\frac{\partial W}{\partial X_2}\right)d\Omega = \int \Psi\phi\,d\Omega \tag{5.15}$$

where ψ is the weight function, $\psi = \omega = \varpi$.

Since the electrostatic field is induced by charges on the particles and the amount of electrostatic charges on each particle is very small, the dimensionless electrostatic charge parameter Q is assumed to range from 0.0001 to 0.005. For example if we use the data employed in section 5.2 for the oil fluid flow with $\phi_o = 0.3$, $u_o = 0.8$ m/s, h = 0.0254 m, d = 250 micro-meters (or d/h = 0.01) and assume an electron per $1.18\text{X}10^{-10}$ cm^2 of particle surface area [42], we have Re = 46, Fr = 0.38, q/mp = 3.3 X 10^{-7} coul/kg and Q = 3.7. If we take 0.1 % of this value and let Q = 0.003 we will have a concentration distribution as shown in one of the following figures. The particle concentration in this case is 37,000 particles per cc and is within the lower limit of concentration for which we can apply the repulsive force approach as is used in this analysis.

The numerical results are shown in figures 5.9 through 5.10 (a-d). Figure 5.9 depicts the electrostatic potential W at Re = 46, Fr = 0.38, ϕ_o = 0.30 and d/h = 0.03. As is seen the potential is highest near the centerline of the channel and decreases to zero at walls. The electrostatic field intensity is equal to – dw/dx_2 and is therefore maximum near the walls. The electrostatic force is equal to the product of electrostatic field intensity and the charges on each particle.

Figure 5.10(a) depicts the effect of Q on particle volume fraction distribution at ϕ_o= 0.3, Re = 46, Fr = 0.38 and d/h=0.01. Figure 5.10(b) depicts the effect of Q on the particle volume fraction distribution at ϕ_o= 0.3, Re = 46, Fr = 0.38 and d/h=0.03. We can see that the particle concentration at the peak area near the top wall will decrease if Q exists. The higher the Q parameter, the lower the particle concentration at the peak area. This is due to the fact that charged particles experience repulsive force toward the walls. Moreover there will appear another concentrated area at the centerline.

Figure 5.10(c) and 5.10(d) show the particle concentration distribution at various Q parameter at Re = 46 (Fr = 0.38) and 18 (Fr = 0.15) respectively for ϕ_o= 0.35, and d/h=0.03. We can see at Re=18, the effect of Q on particle concentration distribution is more pronounced than that at Re = 46. This can be seen from the governing equation (5.9) that the lower Reynolds number produces higher Q effects on the flow.

It should be noted that based on the definition of Q, the parameter is proportional to q^2 but inversely proportional to μ_f^2 Thus for the same amount of charges on a particle the value of Q will decrease if the dynamic viscosity of the fluid is increased. To discern the coupled effects of viscosity and charge on the flow we can exam equation (5.5). It is seen that as the dynamic viscosity of the fluid is increased the charge effects is diminished.

The above results have shown that when the charge parameter is small ($Q<0.001$) the effects of the charge is to reduce the peak concentration near the top wall. It is expected that when $Q>0.001$ the effects of charge may become significant.

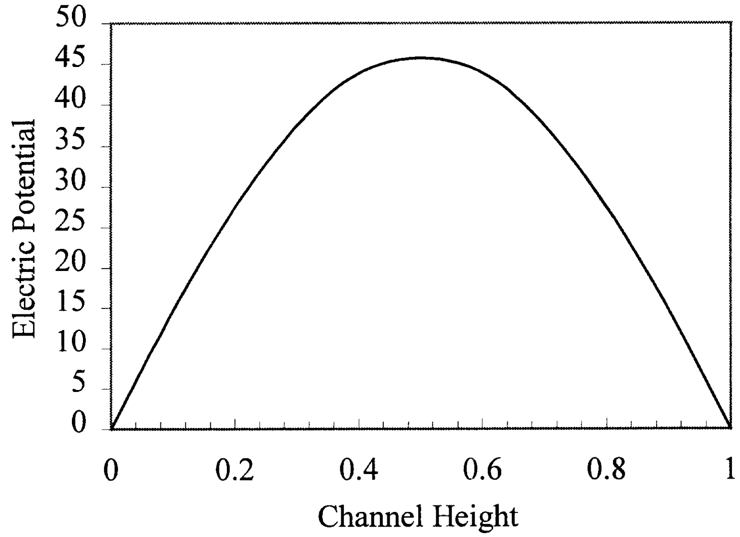


Figure 5.9 Electrostatic potential W at Re = 46, Fr = 0.38, $\phi_o = 0.30$ and d/h = 0.03.

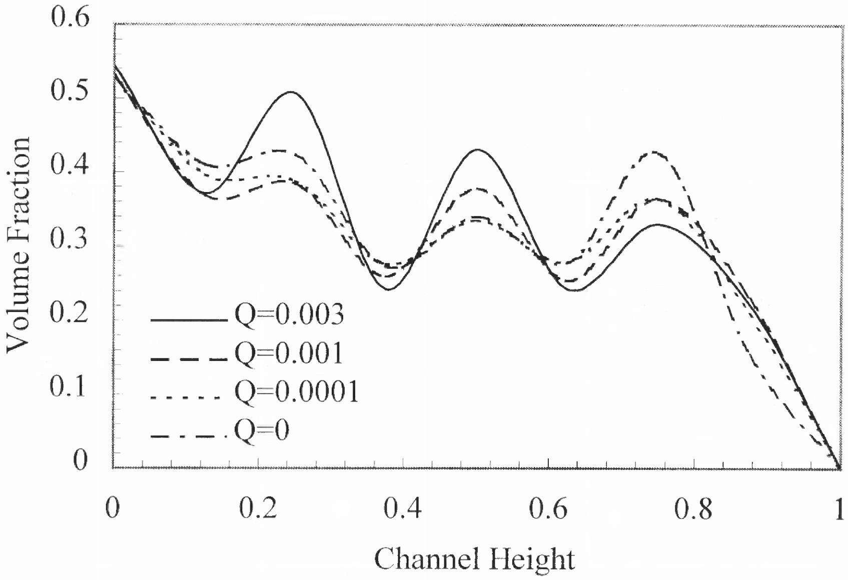


Figure 5.10(a) Effect of Q on the ϕ distribution with ϕ_0 =0.3, Re=46, d/H=0.01.

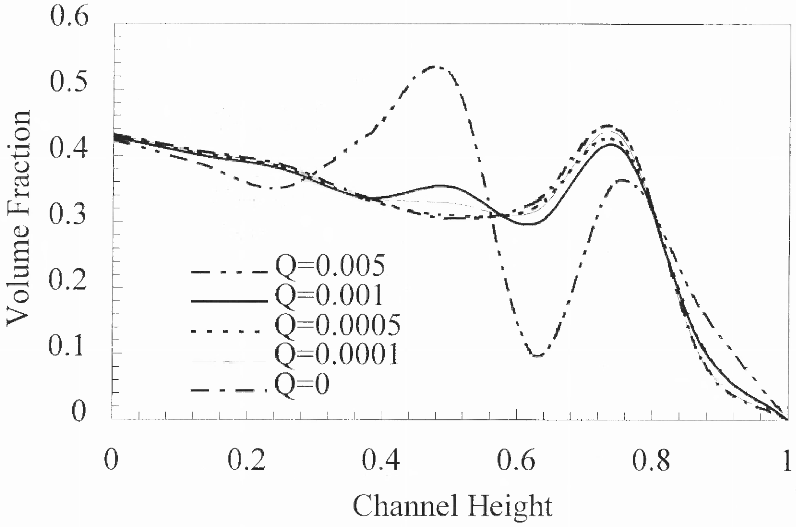


Figure 5.10(b) Effect of Q on the ϕ distribution with ϕ_0 =0.3, Re=46,d/H=0.03.

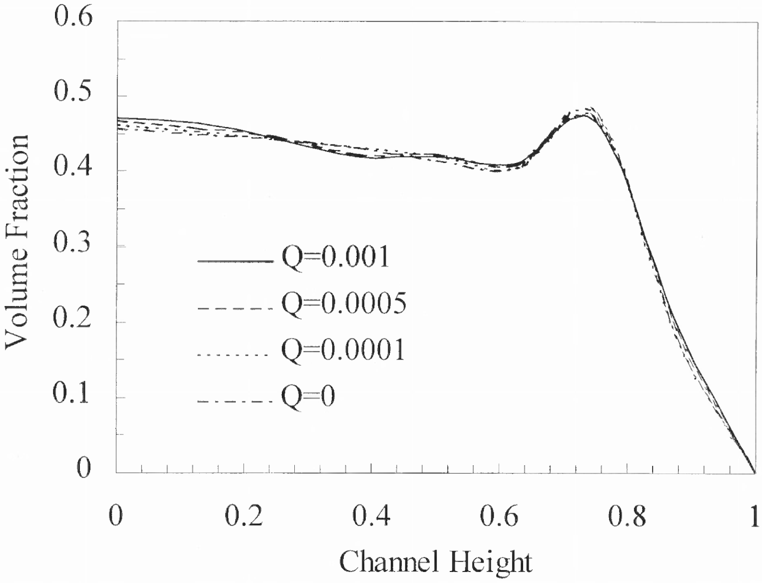


Figure 5.10(c) Effect of Q on the ϕ distribution with ϕ_0 =0.35, Re=46, d/H=0.03.

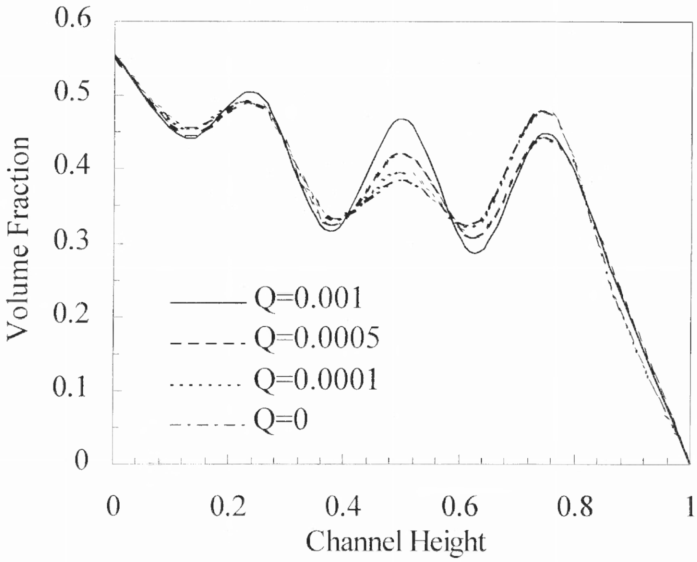


Figure 5.10(d) Effect of Q on the ϕ distribution with ϕ_0 =0.35, Re=18, d/H=0.03.

5.9 Comparison With the Experimental Study

As we mentioned in chapter 2, there are many experimental studies on the viscous suspension flow, most of the experimental results demonstrated that in the channel flow, there will be a particle concentrated region above the centerline, and with the increase of the inlet particle concentration, the concentrated region will become bigger and the maximum particle concentration will be at this concentrated region. The experimental study also found that the axial velocity will become blunt compare with the pure fluid, the higher the inlet particle concentration, the more asymmetric of the velocity profile.

Here we use Altobelli[17] et al's research to compare with our numerical calculations.

Altobelli et al employed the nuclear magnetic resonance imaging technology to measure the velocity and concentration of suspensions. They measured the suspensions of small, negatively buoyant particles flowing in horizontal tubes. All measurements were performed during statistically steady flow conditions with average fluid velocities up to 25cm / s with the mean particle concentration ϕ_0 between $0.0 < \phi_0 < 0.39$. The density of

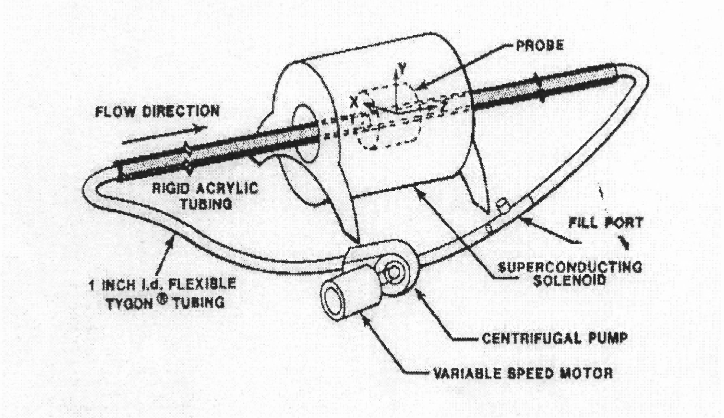


Figure 5.11 Schematic of the flow loop apparatus of Altobelli[17] et al's experiment.

particles are $1003kg/m^3$, and the mean diameter of the particles are 0.762mm. The suspending fluid has the viscosity of 3.84P, and the density of $875kg/\ m^3$. In our study, we use this data as one set of our parameter combinations.

Figure 5.11 shows the schematic of the flow loop apparatus, the suspensions were circulated in this closed loop.

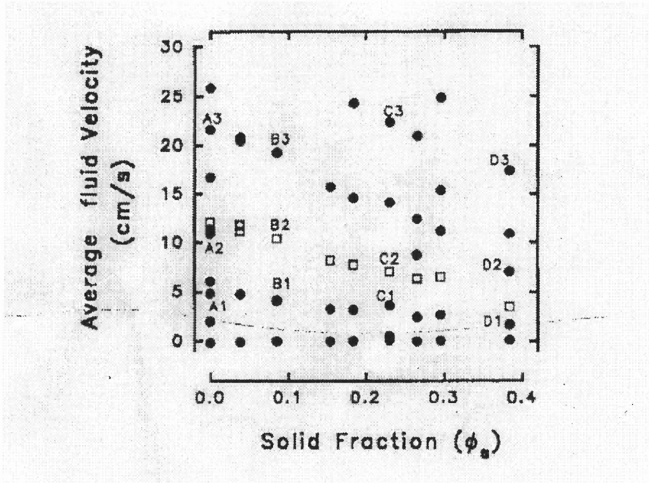


Figure 5.12 Matrix of flow of Altobelli[17] et al's experiments.

Figure 5.12 illustrates the mean fluid velocities and particle volume fractions for their experiments performed, and figure 5.113 shows the results of their experiments. Figure 5.13 contains 12 experimental conditions. A pair of false-color images is shown for each condition. The upper image shows the particle concentration, and the lower image shows the axial velocities. In their results, we can see a region of clear fluid flows above a highly concentrated bed. The thickness of the clear fluid layer decreases as ϕ_0 increases. And even at low flow rate, the particle bed is observed to dilate and consequently become resuspended. They also observed some circular region centered above the tube centerline. These regions are local maximum in the particle concentration

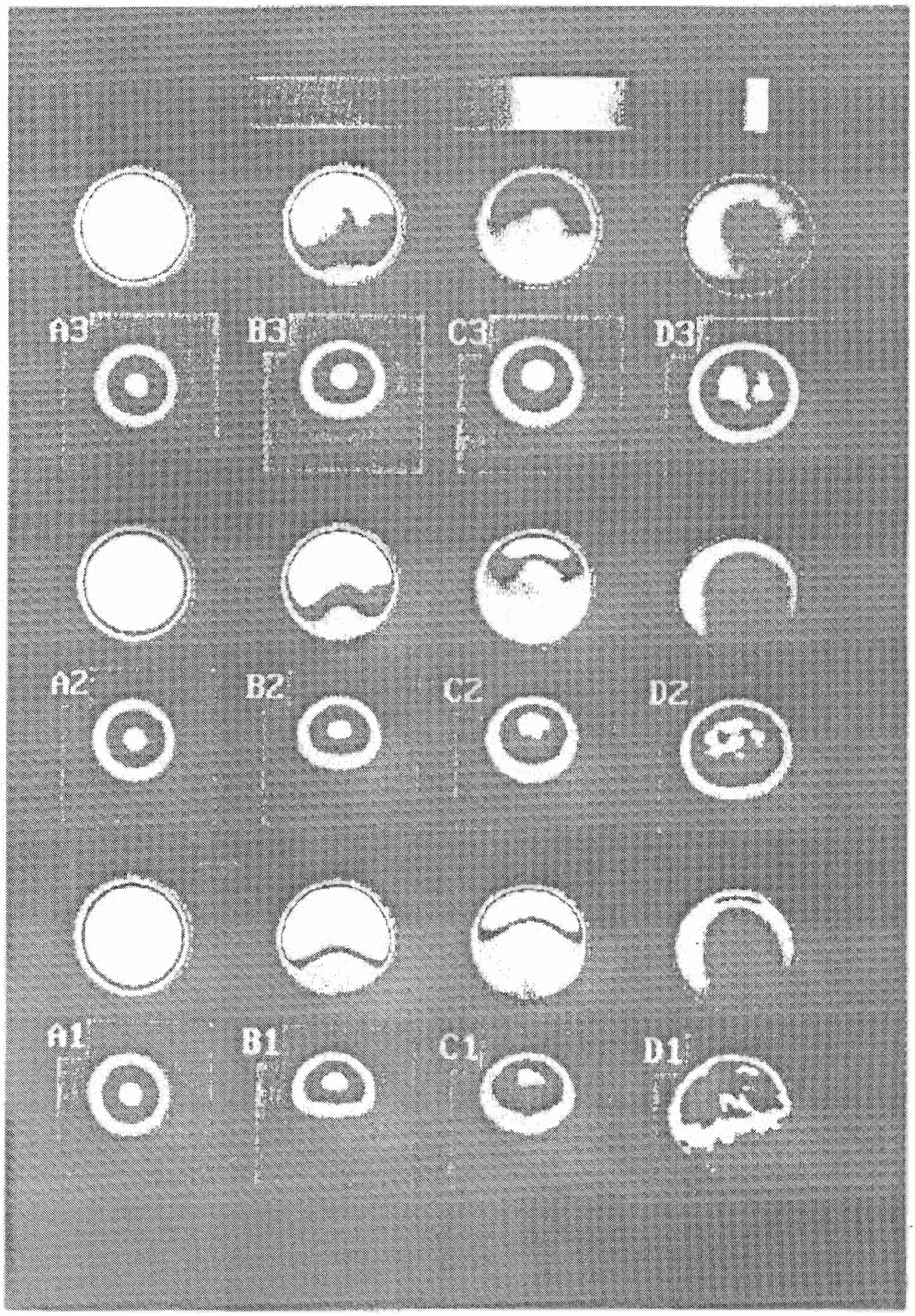


Figure 5.13 Altobelli[17] et al's experimental results of non-homogeneous distribution of axial fluid velocity and particle concentration for negatively buoyant suspensions flowing in horizontal tubes

and the particle concentration in these region is approximately equal to the maximum packing fraction of particles measured in the static samples. Velocity profiles are blunt at high inlet particle concentration and shifted slightly upward.

Since their results were plotted in false color image, we can’t compare their results with our calculation results in quantity, but all the phenomenon they observed in their experiments can be found in our calculation results.

CHAPTER 6

CONCLUSION

When a well mixed concentrated suspension enters a two-dimensional channel the suspended particles migrate within the flow. This migration is caused by three factors within the flow; shear gradients, relative viscosity gradients and concentration gradients. The other forces that may affect the concentration distribution are the gravity effect and the electrostatic effect. In this study the parameters included are the ratio of particle diameter to channel height d/H, Reynolds number Re, Froude number Fr, uniform inlet particle volume fraction ϕ_o, and electrostatic parameter Q.

The following conclusions are based on the analysis of above parameters.

1. Under the effect of gravity, there is a dilute concentration region near the top wall. The concentration is zero at the top wall and is maximum at the bottom wall.
2. For any given particle material the concentration at the bottom wall increases with decreasing mean flow velocity and increases with decreasing particle size.
3. There is a maximum concentration region half way between the top wall and the centerline. This maximum concentration increases with decreasing d/H ratio. When the viscous induced diffusion effect is excluded the concentration profile increases monotonically and does not show this maximum concentration.
4. The concentration in the centerline region also exhibit a maximum when $d/H \leq 0.06$ and $\phi_o \leq 0.30$. The concentration is about 10 % higher than that at the inlet.
5. Another concentration peak is in the region between the centerline and the bottom wall ($X_2=0.25$). The peak concentration rages between 1.5 ϕ_o to 1.1ϕ_o.

6. The maximum concentration near the top wall increases with increasing ϕ_o for $0.3>\phi_o>0.20$. At $d/H=0.03$ and $\phi_o=0.30$ this maximum is $\phi/\phi_o = 1.40$.
7. Velocity profile deviate from the fully-developed parabolic profile in the top half of the channel. This increase in velocity is higher with smaller d/H, higher ϕ_o and higher Re.
8. The effect of electrostatic charge on the concentration profile is negligible for $Q<0.0001$ and becomes more significant as the value of Q is increased over 0.001.
9. Comparison of some experimental results on tube flow gives qualitatively good agreement.
10. Analysis on developing flow shows that the entrance length for concentration is also a function of inlet concentration.

The following suggestion is made to be the direction of future research on this subject

1. We may extend the study to three-dimension analysis.
2. In our study, at the suspension-pure fluid surface, we assume that if the particle concentration is less than 0.002, it will treated as 0. The future study can use more accurate method (stream-up wind / Petrov-Galerkin method, Taloy-Galerkin method) to deal with the sharp variation problem.
3. In our study, the effect of electric field is very limited. One may make more study on this aspect.

APPENDIX

C++ PROGRAM FOR DEVELOPING FLOW WITHOUT ELECTROSTATIC CHARGE

```
#include <math.h>
#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <stdio.h>
class Elements
{
public:
              int node[9];
              double x_coordinate [9];
              double y_coordinate [9];
} ;
class one_element
{
public:
      int ndnumber[9];
      double psai_gauss [9];
              double u_gauss [9];
              double v_gauss [9];
              double dxu_gauss [9];
              double dyu_gauss [9];
              double dxv_gauss [9];
              double dyv_gauss [9];
              double dxxu_gauss [9];
              double dyyu_gauss [9];
              double dxxv_gauss [9];
              double dyyv_gauss [9];
              double dxyu_gauss [9];
              double dxyv_gauss [9];
              double dxpsai_gauss [9];
              double dypsai_gauss [9];
              double Dc [9];
              double Ds [9];
              double gamadot [9];
              double dx_gamadot [9];
              double dy_gamadot [9];
              int node[9];
              double xcoordinate [9];
              double ycoordinate [9];
              double x_velocity  [9];
              double y_velocity  [9];
              double psai [9];
              double S11[9][9], S22[9][9], S33[9][9];
              double M11[9][9], M22[9][9], M33[9][9];
              double L11[9][9],L12[9][9],L21[9][9],L22[9][9];
              double F1[9],F11[9],F22[9],F2[9],F3[9], F33[9], F[18];
              double AE1[9][9], AE2[18][18],AE12[18][18];
              double Deltat;
              double Global1[711][21], Global2[1422][42];
              double GF1[711], GF2[1422];
              void  define_0();
              void cal_AE1( );
              void define_matrix0( );
              void cal_F( );
              void cal_global1(int nf);
              void define_matrix02();
              void define_matrix03();
```

```
              void define_02();
              void cal_AE12( );
              void cal_AE2( );
              void cal_F1( );
              void cal_F2( );
              void cal_MF( );
              void cal_global2( int nf);
};

void calnode(int elenode[9], int x,int y, int ty);
void cal_coorx(double x[9], double fxx[2][11], int ii, int jj, int nxx);
void cal_coory(double y[9], double fyy[2][81], int i2, int j2, int nyy);
void cal_fai(double fai[9],double att, double kcc);
void cal_kc1(double d1kc[9], double att, double kcc);
void cal_kc2(double d2kc[9], double att, double kcc);
void cal_at1(double d1at[9], double att, double kcc);
void cal_at2(double d2at[9], double att, double kcc);
void cal_kcat(double dkcat[9], double att, double kcc);
void cal_J(double array1[2][2], double a1[9], double b1[9], double
c1[9], double d1[9]);
void cal_invJ(double inJ[2][2], double array2[2][2]);
void cal_J1(double array2[3][3], double array1[2][2]);
void cal_invJ1(double array2[3][3], double array1[3][3]);
void cal_J2(double arrar1[3][2], double a1[9], double b1[9],double
ab[9], double c1[9], double d1[9]);
void boundary1(double ag[711][21], double bf[711], int cny, int
dnodenumber, int ehbw1, int fnf1);
void solve1(double ag[711][21], double bf[711], int cnodenumber, int
dhbw1);
void boundary2(double ag[1422][42], double bf[1422], int cnx, int cny,
int dnodenumber, int ehbw2, int fnf2);
void solve2(double ag[1422][42], double bf[1422], int cnodenumber, int
dhbw2);
int main( )
{
      Elements eleme [156]  ;
      one_element anelement;
      int *p;
      double *p1;
      double *p2;
      p=(int *)(calloc(9, sizeof(int)));
      p1=(double *)(calloc(9, sizeof(double)));
      p2=(double *)(calloc(9, sizeof(double)));
      int nx=40;
      int ny=5;
      int nf1=1;
      int nf2=2;
      // Material property;
      double H1=0.0254;
      double L=40.0 ;
      double H=1;
      double rou1=875;
      double rou2=1003;
      double Mu1=0.384;
      double U0=0.48;
      double d=0.000762;
      double alfa=1e8;
```

```
// element information;
int HBW1=0;
int HBW2=0;
int NDF1=1;
int NDF2=2;
int elementnumber=(nx-1)*(ny-1);
int nodenumber=(2*nx-1)*(2*ny-1);
int NEQ1=nodenumber*NDF1;
int NEQ2=nodenumber*NDF2;
// Paramters;
double Re=rou1*U0*H1/Mu1;
double Fr=Mu1*U0/(rou2-rou1)/9.81/H1/H1;
double Lamda=(d/H1)*(d/H1);
double epslon=(rou2-rou1)/rou1;
//calculate node index;
int ele=-1;
for(int lnx=1;lnx<=(nx-1);lnx++)
{
      for (int lny=1; lny<=(ny-1);lny++)
      {
            ele=ele++;
            calnode(p,lnx,lny, ny);
            for(int j=0;j<9;j++)
            {
                  eleme[ele].node[j]=p[j];}
      }
}
double fx[2][11],fy[2][81];
for (int ll=1;ll<=2*ny-1;ll++)
{
      fx[0][ll-1]=L;
      fx[1][ll-1]=0; }
for (int mm=1;mm<=2*nx-1;mm++)
{
      fy[0][mm-1]=H;
      fy[1][mm-1]=0; }
//calculate node coordinates;
int ele1=-1;
for (int i=2;i<=2*nx-1;i=i+2)
{
      for (int j=2; j<=2*ny-1;j=j+2)
      {
            ele1=ele1++;
            cal_coorx(p1, fx,i,j, nx);
            for(int j1=0; j1<9; j1++)
            {
                  eleme[ele1].x_coordinate[j1]=p1[j1];
            }
            cal_coory(p2, fy,i,j, ny);
            for (int j11=0; j11<9; j11++)
            {
                  eleme[ele1].y_coordinate[j11]=p2[j11];
                  //cout<<eleme[ele1].y_coordinate[j11]<<" ";
            }
      }
}
// Calculate the half band width;
```

```
	int nw1, nw2;
	for (int ee=0; ee<elementnumber; ee++)
	{
		for (int i=0;i<9;i++)
		{
			for (int j=0; j<9;j++)
			{
				//cout<<eleme[ee].node[i]<<"
"<<eleme[ee].node[j]<<endl;
				nw1=(abs(eleme[ee].node[i]-
eleme[ee].node[j])+1)*NDF1;
				nw2=(abs(eleme[ee].node[i]-
eleme[ee].node[j])+1)*NDF2;
				if (HBW1<nw1)
				{
					HBW1=nw1;}
				if (HBW2<nw2)
				{
					HBW2=nw2;}
			}
		}
	}
	//cout<<HBW1<<" "<<HBW2<<endl;
	double Ux[800], Vy[800], Cfai[800];
	//initial value of U, V and fai;
	for ( i=0; i<nodenumber; i++)
	{
		if ((i+1)%(2*ny-1)==0 || ((i+1)%(2*ny-1)==1))
		{
			Ux[i]=0;
			Vy[i]=0;
		}
		else
		{
			Ux[i]=1.0;
			Vy[i]=0;
		}
		Cfai[i]=0.30
			;
		//cout<<i<<" "<<Ux[i]<<" ";
	}
	double time=0.0;
	anelement.Deltat=0.01;
	for (int Tnumber=0;Tnumber<200; Tnumber++)
	{
		for (int ii=0; ii<nodenumber*NDF1; ii++)
		{
			for (int jj=0; jj<HBW1; jj++)
			{
				anelement.Global1[ii][jj]=0;}
			anelement.GF1[ii]=0;
		}
		for (int iii=0; iii<nodenumber*NDF2; iii++)
		{
			for (int jj=0; jj<HBW2; jj++)
			{
				anelement.Global2[iii][jj]=0;}
```

```
            anelement.GF2[iii]=0;
      }
      for (int el=0; el<elementnumber; el++)
      {
            for (int ll=0; ll<9; ll++)
            {
                  int nn=eleme[el].node[ll];
                  //cout<<el<<eleme[el].y_coordinate[ll]<<" ";
                  anelement.ndnumber[ll]=nn-1;
                  anelement.x_velocity[ll]=Ux[nn-1];
                  anelement.y_velocity[ll]=Vy[nn-1];
            //cout<<(nn-1)<<" "<<anelement.x_velocity[ll]<<" ";
                  anelement.psai[ll]=Cfai[nn-1];
            }
      //calculate the velocity and volume factor of Gauss Point;
            double *ai, *kc1, *kc2, *at1, *at2, *kcat;
            ai=(double *)(calloc(9, sizeof(double)));
            kc1=(double *)(calloc(9, sizeof(double)));
            kc2=(double *)(calloc(9, sizeof(double)));
            at1=(double *)(calloc(9, sizeof(double)));
            at2=(double *)(calloc(9, sizeof(double)));
            kcat=(double *)(calloc(9, sizeof(double)));
            double c=0.77459667;
            int ii=-1;
            anelement.define_0();
            double kc=-c;
            for (int i=0; i<3;i++)
            {
                  double at=-c;
                  for (int j=0; j<3;j++)
                  {
                        ii=ii+1;
                        cal_fai(ai, at,kc);
                        cal_kc1(kc1, at,kc);
                        cal_kc2(kc2, at,kc);
                        cal_at1(at1, at,kc);
                        cal_at2(at2, at,kc);
                        cal_kcat(kcat, at,kc);
      double J[2][2], invJ[2][2], J1[3][3], invJ1[3][3], J2[3][2];
                        for (int k1=0; k1<2; k1++)
                        {
                              for (int k2=0; k2<2; k2++)
                              {
                                    J[k1][k2]=0;
                                    invJ[k1][k2]=0;}
                        }
cal_J(J,kc1, at1, eleme[el].x_coordinate, eleme[el].y_coordinate);
      //cout<<J[0][0]<<" "<<J[0][1]<<" "<<J[1][0]<<" "<<J[1][1];
                        cal_invJ(invJ,J);
                        for(int kk1=0; kk1<3; kk1++)
                        {
                              for (int kk2=0; kk2<3; kk2++)
                              {
                                    J1[kk1][kk2]=0;}
                        }
                        cal_J1(J1, J);
                        cal_invJ1(invJ1, J1);
```

```
                              for(int k11=0; k11<3; k11++)
                              {
                                    for (int k22=0; k22<2; k22++)
                                    {
                                          J2[k11][k22]=0;}
                              }
cal_J2(J2,kc2,at2, kcat,eleme[el].x_coordinate, eleme[el].y_coordinate);
                              double ppfx[9], ppfy[9], J3[3];
                              double p2fx[9], p2fy[9], pfxfy[9];
                              for(int k=0; k<9;k++)
                              {

      ppfx[k]=(invJ[0][0]*kc1[k]+invJ[0][1]*at1[k]);

      ppfy[k]=(invJ[1][0]*kc1[k]+invJ[1][1]*at1[k]);
            J3[0]=kc2[k]-J2[0][0]*ppfx[k]-J2[0][1]*ppfy[k];
            J3[1]=kc2[k]-J2[1][0]*ppfx[k]-J2[1][1]*ppfy[k];
            J3[2]=kc2[k]-J2[2][0]*ppfx[k]-J2[2][1]*ppfy[k];

      p2fx[k]=invJ1[0][0]*J3[0]+invJ1[0][1]*J3[1]+invJ1[0][2]*J3[2];

      p2fy[k]=invJ1[1][0]*J3[0]+invJ1[1][1]*J3[1]+invJ1[1][2]*J3[2];

      pfxfy[k]=invJ1[2][0]*J3[0]+invJ1[2][1]*J3[1]+invJ1[2][2]*J3[2];
anelement.psai_gauss[ii]=anelement.psai_gauss[ii]+anelement.psai[k]*ai[k
];

      anelement.dxpsai_gauss[ii]=anelement.dxpsai_gauss[ii]+
      anelement.psai[k]*ppfx[k];

      anelement.dypsai_gauss[ii]=anelement.dypsai_gauss[ii]+
      anelement.psai[k]*ppfy[k];
anelement.u_gauss[ii]=anelement.u_gauss[ii]+anelement.x_velocity[k]*ai[k
];

      anelement.dxu_gauss[ii]=anelement.dxu_gauss[ii]+
      anelement.x_velocity[k]*ppfx[k];
anelement.v_gauss[ii]=anelement.v_gauss[ii]+anelement.y_velocity[k]*ai[k
];

      anelement.dyu_gauss[ii]=anelement.dyu_gauss[ii]+
      anelement.x_velocity[k]*ppfy[k];

      anelement.dxv_gauss[ii]=anelement.dxv_gauss[ii]+
      anelement.y_velocity[k]*ppfx[k];

      anelement.dyv_gauss[ii]=anelement.dyv_gauss[ii]+
      anelement.y_velocity[k]*ppfy[k];

      anelement.dxxu_gauss[ii]=anelement.dxxu_gauss[ii]+
      anelement.x_velocity[k]*p2fx[k];

      anelement.dxxv_gauss[ii]=anelement.dxxv_gauss[ii]+
      anelement.y_velocity[k]*p2fx[k];

      anelement.dyyu_gauss[ii]=anelement.dyyu_gauss[ii]+
      anelement.x_velocity[k]*p2fy[k];
```

```
 anelement.dyyv_gauss[ii]=anelement.dyyv_gauss[ii]+
 anelement.y_velocity[k]*p2fy[k];

 anelement.dxyu_gauss[ii]=anelement.dxyu_gauss[ii]+
 anelement.x_velocity[k]*pfxfy[k];

 anelement.dxyv_gauss[ii]=anelement.dxyv_gauss[ii]+
 anelement.y_velocity[k]*pfxfy[k];
                         }

 anelement.gamadot[ii]=sqrt(2.*pow(anelement.dxu_gauss[ii],2.)+
 2.*pow(anelement.dyv_gauss[ii],2.)+pow((anelement.dyu_gauss[ii]+
 anelement.dxv_gauss[ii]),2.));
                         if (anelement.gamadot[ii]<1e-5)
                         {
                               anelement.gamadot[ii]=1e-5;}

 anelement.dx_gamadot[ii]=pow(anelement.gamadot[ii],-0.5)*
(4.*anelement.dxu_gauss[ii]*anelement.dxxu_gauss[ii]+
4.*anelement.dyv_gauss[ii]*anelement.dxyv_gauss[ii]+
2.*(anelement.dyu_gauss[ii]+anelement.dxv_gauss[ii])*
(anelement.dxyu_gauss[ii]+anelement.dxxv_gauss[ii]))/2;

 anelement.dy_gamadot[ii]=pow(anelement.gamadot[ii],-0.5)*
(4.*anelement.dxu_gauss[ii]*anelement.dxyu_gauss[ii]+
4.*anelement.dyv_gauss[ii]*anelement.dyyv_gauss[ii]+
2.*(anelement.dyu_gauss[ii]+anelement.dxv_gauss[ii])*
(anelement.dyyu_gauss[ii]+anelement.dxyv_gauss[ii]))/2;


 anelement.Dc[ii]=1.0/3.0*pow(anelement.psai_gauss[ii],2.)*
 (1+0.5*exp(8.8*anelement.psai_gauss[ii]));

 anelement.Ds[ii]=0.6*pow(anelement.psai_gauss[ii],2.);
                         at=at+c;}
                   kc=kc+c;}
 double rou=1.+epslon*anelement.psai_gauss[4];
 double Mu=pow((1.+(1.5*anelement.psai_gauss[4]/
(1.-anelement.psai_gauss[4]/0.58))),2.);
             double f=(1.-anelement.psai_gauss[4])/Mu;
             //calculate matrix S33 and M33;
             anelement.define_matrix0( );
             double pfx[9], pfy[9];
             double delJ;
             for (int m=0; m<9; m++)
             {
                   for (int n=0; n<9; n++)
                   {
                         int ii=-1;
                         double kc=-c;
                         for (int i=0; i<3; i++)
                         {
                               double at=-c;
                               for(int j=0; j<3; j++)
                               {
                                     ii=ii+1;
```

```
                                          cal_fai(ai, at,kc);
                                          cal_kc1(kc1, at,kc);
                                          cal_at1(at1, at,kc);
                                          double J[2][2], invJ[2][2];
                                          for (int k1=0; k1<2; k1++)
                                          {
                                          for (int k2=0; k2<2; k2++)
                                                {
                                                      J[k1][k2]=0;}
                                                //invJ[k1][k2]=0;}
                                          }
      cal_J(J,kc1,at1, eleme[el].x_coordinate, eleme[el].y_coordinate);
            cal_invJ(invJ,J);
            delJ=J[0][0]*J[1][1]-J[0][1]*J[1][0];

      pfx[m]=(invJ[0][0]*kc1[m]+invJ[0][1]*at1[m]);

      pfx[n]=(invJ[0][0]*kc1[n]+invJ[0][1]*at1[n]);

      pfy[m]=(invJ[1][0]*kc1[m]+invJ[1][1]*at1[m]);

      pfy[n]=(invJ[1][0]*kc1[n]+invJ[1][1]*at1[n]);
                                          double WI, WJ;
                                          if(kc==0)
                                                WI=0.8888888889;
                                          else    WI=0.5555555555;
                                          if(at==0)
                                                WJ=0.8888888889;
                                          else   WJ=0.5555555555;
anelement.S33[m][n]=anelement.S33[m][n]+(pfx[m]*pfx[n]+pfy[m]*pfy[n])*
anelement.Dc[ii]*anelement.gamadot[ii]*Lamda*WI*WJ*delJ*Mu1*Re*rou;

      anelement.M33[m][n]=anelement.M33[m][n]+ai[m]*ai[n]*WI*WJ*delJ;
                                    at=at+c;}

                                    kc=kc+c;
                              }
                        }
                  }
                  double pfx1[9], pfy1[9];
                  double detJ;
                  for (int mm=0; mm<9; mm++)
                  {
                        int ii=-1;
                        double kc=-c;
                        for (int i=0; i<3; i++)
                        {
                              double at=-c;
                              for(int j=0; j<3; j++)
                              {
                                    ii=ii+1;
                                    cal_fai(ai, at,kc);
                                    cal_kc1(kc1, at,kc);
                                    cal_at1(at1, at,kc);
                                    double J[2][2], invJ[2][2];
                                    for (int k1=0; k1<2; k1++)
                                    {
```

```
                            for (int k2=0; k2<2; k2++)
                            {
                                J[k1][k2]=0;}
                            //invJ[k1][k2]=0;}
                        }
    cal_J(J,kc1,at1, eleme[el].x_coordinate, eleme[el].y_coordinate);
        cal_invJ(invJ,J);
        detJ=J[0][0]*J[1][1]-J[0][1]*J[1][0];

    pfx1[mm]=(invJ[0][0]*kc1[mm]+invJ[0][1]*at1[mm]);

    pfy1[mm]=(invJ[1][0]*kc1[mm]+invJ[1][1]*at1[mm]);
                        double WI, WJ;
                        if(kc==0)
                            WI=0.8888888889;
                        else   WI=0.5555555555;
                        if(at==0)
                            WJ=0.8888888889;
                        else  WJ=0.5555555555;

anelement.F33[mm]=anelement.F33[mm]-anelement.Deltat*
(ai[mm]*(anelement.u_gauss[ii]*anelement.dxpsai_gauss[ii]+anelement.v_ga
uss[ii]*anelement.dypsai_gauss[ii])+Lamda*2./9.*pfy1[mm]*f*anelement.psa
i_gauss[ii]/Fr)*WI*WJ*detJ;
anelement.F33[mm]=anelement.F33[mm]-anelement.Deltat*
(2./9.*pfy1[mm]*f*anelement.psai_gauss[ii]/Fr)*WI*WJ*detJ;


                        at=at+c;
                    }
                    kc=kc+c;
                }
            }
            //for (int aa=0; aa<9; aa++)
            //{cout<<el<<" "<<anelement.F33[aa]<<endl;}
            anelement.cal_F( );
            anelement.cal_AE1();
            anelement.cal_global1(nf1);
}
boundary1(anelement.Global1, anelement.GF1, ny, nodenumber, HBW1, NDF1);
solve1(anelement.Global1, anelement.GF1, nodenumber, HBW1);
for (int lmn=0; lmn<nodenumber; lmn++)
{
    Cfai[lmn]=anelement.GF1[lmn];
    if (Cfai[lmn]<0.002)
    {
        Cfai[lmn]=0;}
}
fstream myfile;
myfile.open("psai.dat", ios::app);
for(int im=0; im<nodenumber;im++)
{if (Tnumber%2==1)
    myfile<<" "<<Cfai[im]<<" "<<endl;}
myfile.close();
for (int el2=0; el2<elementnumber;el2++)
{
    anelement.define_matrix02();
```

```
      for (int ll=0; ll<9; ll++)
      {
            int nn=eleme[el2].node[ll];
            anelement.ndnumber[ll]=nn-1;
            anelement.x_velocity[ll]=Ux[nn-1];
            anelement.y_velocity[ll]=Vy[nn-1];
            anelement.psai[ll]=Cfai[nn-1];
            //cout<<(nn-1)<<" "<<anelement.psai[ll]<<" ";
      }
      double c2=0.77459667;
      double *ai, *kc1, *kc2, *at1, *at2, *kcat;
      ai=(double *)(calloc(9, sizeof(double)));
      kc1=(double *)(calloc(9, sizeof(double)));
      kc2=(double *)(calloc(9, sizeof(double)));
      at1=(double *)(calloc(9, sizeof(double)));
      at2=(double *)(calloc(9, sizeof(double)));
      kcat=(double *)(calloc(9, sizeof(double)));
      int ii2=-1;
      anelement.define_02();
      double kct=-c2;
      for (int in=0; in<3; in++)
      {
            double att=-c2;
            for(int jn=0; jn<3; jn++)
            {
                  ii2=ii2+1;
                  cal_fai(ai, att, kct);
                  cal_kc1(kc1, att, kct);
                  cal_at1(at1, att, kct);
                  double J[2][2], invJ[2][2];
                  for (int k1=0; k1<2; k1++)
                  {
                        for (int k2=0; k2<2; k2++)
                        {
                              J[k1][k2]=0;
                        }
                        //invJ[k1][k2]=0;}
                  }
cal_J(J,kc1,at1, eleme[el2].x_coordinate, eleme[el2].y_coordinate);
      // cout<<J[0][0]<<" "<<J[0][1]<<" "<<J[1][0]<<" "<<J[1][1]<<endl;
                  cal_invJ(invJ,J);
                  double ppfx[9], ppfy[9];
                  for(int k=0; k<9;k++)
                  {
                        ppfx[k]=(invJ[0][0]*kc1[k]+invJ[0][1]*at1[k]);
                        ppfy[k]=(invJ[1][0]*kc1[k]+invJ[1][1]*at1[k]);
                        //cout<<ppfx[k]<<" "<<ppfy[k]<<endl;
anelement.psai_gauss[ii2=anelement.psai_gauss[ii2]+anelement.psai[k]*ai[
k];
anelement.dxpsai_gauss[ii2]=anelement.dxpsai_gauss[ii2]+
anelement.psai[k]*ppfx[k];

anelement.dypsai_gauss[ii2]=anelement.dypsai_gauss[ii2]+
anelement.psai[k]*ppfy[k];

anelement.u_gauss[ii2]=anelement.u_gauss[ii2]+anelement.x_velocity[k]*ai
[k];
```

```
anelement.dxu_gauss[ii2]=anelement.dxu_gauss[ii2]+
anelement.x_velocity[k]*ppfx[k];

anelement.v_gauss[ii2]=anelement.v_gauss[ii2]+anelement.y_velocity[k]*ai
[k];
anelement.dyu_gauss[ii2]=anelement.dyu_gauss[ii2]+
anelement.x_velocity[k]*ppfy[k];

anelement.dxv_gauss[ii2]=anelement.dxv_gauss[ii2]+
anelement.y_velocity[k]*ppfx[k];

anelement.dyv_gauss[ii2]=anelement.dyv_gauss[ii2]+
anelement.y_velocity[k]*ppfy[k];
                  }
                  att=att+c2;}
            kct=kct+c2;}
double rou=1+ epslon*anelement.psai_gauss[4];
double    Mu=pow((1.+(1.5*anelement.psai_gauss[4]/
(1.-anelement.psai_gauss[4]/0.58))),2.);
      double      f=(1.-anelement.psai_gauss[4])/Mu;
      //cout<<rou<<" "<<Mu<<" "<<f<<endl;
      //calculate matrix S11,S22 and M11,M22;
      anelement.define_matrix0( );
      double pfxv[9], pfyv[9];
      double delJv;
      for (int m=0; m<9; m++)
      {
            for (int n=0; n<9; n++)
            {
                  int ii=-1;
                  double kc=-c2;
                  for (int i=0; i<3; i++)
                  {
                        double at=-c2;
                        for(int j=0; j<3; j++)
                        {
                              ii=ii+1;
                              cal_fai(ai, at,kc);
                              cal_kc1(kc1, at,kc);
                              cal_at1(at1, at,kc);
                              double J[2][2], invJ[2][2];
                              for (int k1=0; k1<2; k1++)
                              {
                                    for (int k2=0; k2<2; k2++)
                                    {
                                          J[k1][k2]=0;
                                    }
                                    //invJ[k1][k2]=0;
                              }
cal_J(J,kc1,at1, eleme[el2].x_coordinate, eleme[el2].y_coordinate);
            cal_invJ(invJ,J);
            delJv=J[0][0]*J[1][1]-J[0][1]*J[1][0];

      pfxv[m]=(invJ[0][0]*kc1[m]+invJ[0][1]*at1[m]);

      pfxv[n]=(invJ[0][0]*kc1[n]+invJ[0][1]*at1[n]);
```

```
      pfyv[m]=(invJ[1][0]*kc1[m]+invJ[1][1]*at1[m]);

      pfyv[n]=(invJ[1][0]*kc1[n]+invJ[1][1]*at1[n]);
                              double WI, WJ;
                              if(kc==0)
                                    WI=0.8888888889;
                              else    WI=0.5555555555;
                              if(at==0)
                                    WJ=0.8888888889;
                              else  WJ=0.5555555555;
 anelement.S11[m][n]=anelement.S11[m][n]+
(pfxv[m]*pfxv[n]+pfyv[m]*pfyv[n])*Mu*WI*WJ*delJv;
 anelement.M11[m][n]=anelement.M11[m][n]+ai[m]*ai[n]*WI*WJ*delJv*Re*rou;

anelement.S22[m][n]=anelement.S22[m][n]+(pfxv[m]*pfxv[n]+pfyv[m]*pfyv[n]
)*
 Mu*WI*WJ*delJv;
 anelement.M22[m][n]=anelement.M22[m][n]+ai[m]*ai[n]*WI*WJ*delJv*Re*rou;
                              at=at+c2;}
                        kc=kc+c2;
                  }
            }
      }
      double pfxv1[9], pfyv1[9];
      double detvJ;
      for (int mn=0; mn<9; mn++)
      {
            int ii=-1;
            double kc=-c2;
            for (int i=0; i<3; i++)
            {
                  double at=-c2;
                  for(int j=0; j<3; j++)
                  {
                        ii=ii+1;
                        cal_fai(ai, at,kc);
                        cal_kc1(kc1, at,kc);
                        cal_at1(at1, at,kc);
                        double J[2][2], invJ[2][2];
                        for (int k1=0; k1<2; k1++)
                        {
                              for (int k2=0; k2<2; k2++)
                              {
                                    J[k1][k2]=0;}
                              //invJ[k1][k2]=0;}
                        }
cal_J(J,kc1,at1, eleme[el2].x_coordinate, eleme[el2].y_coordinate);
cal_invJ(invJ,J);
detvJ=J[0][0]*J[1][1]-J[0][1]*J[1][0];

      pfxv1[mn]=(invJ[0][0]*kc1[mn]+invJ[0][1]*at1[mn]);

      pfyv1[mn]=(invJ[1][0]*kc1[mn]+invJ[1][1]*at1[mn]);
                        double WI,WJ;
                        if(kc==0)
                              WI=0.8888888889;
```

```
                        else    WI=0.5555555555;
                        if(at==0)
                              WJ=0.8888888889;
                        else  WJ=0.5555555555;

anelement.F11[mn]=anelement.F11[mn]-
anelement.Deltat*ai[mn]*Re*rou*(anelement.u_gauss[ii]*anelement.dxu_gaus
s[ii]+
anelement.v_gauss[ii]*anelement.dyu_gauss[ii])*WI*WJ*detvJ;

anelement.F22[mn]=anelement.F22[mn]-anelement.Deltat*
(ai[mn]*Re*rou*(anelement.u_gauss[ii]*anelement.dxv_gauss[ii]+
anelement.v_gauss[ii]*anelement.dyv_gauss[ii])-
anelement.psai_gauss[ii]*ai[mn]/Fr)*WI*WJ*detvJ;

                        at=at+c2;
                  }
                  kc=kc+c2;
            }
      }
      //for (int bb=0; bb<9; bb++)
      //     { cout<<anelement.F11[bb]<<" "<<anelement.F22[bb]<<endl;}
      anelement.cal_F1( );
      anelement.cal_F2( );
      anelement.define_matrix03();
      double pfxp[9], pfyp[9];
      double detJp;
      double cp=0.5773507;
      for (int mx=0; mx<9; mx++)
      {
            for (int n=0; n<9; n++)
            {
                  double kc=-cp;
                  for (int i=0; i<2; i++)
                  {
                        double at=-cp;
                        for(int j=0; j<2; j++)
                        {
                              cal_fai(ai, at,kc);
                              cal_kc1(kc1, at,kc);
                              cal_at1(at1, at,kc);
                              double J[2][2], invJ[2][2];
                              for (int k1=0; k1<2; k1++)
                              {
                                    for (int k2=0; k2<2; k2++)
                                    {
                                          J[k1][k2]=0;
                                    }
                                    //invJ[k1][k2]=0;}
                              }
cal_J(J,kc1,at1, eleme[el2].x_coordinate, eleme[el2].y_coordinate);
cal_invJ(invJ,J);
//cout<<invJ[0][0]<<" "<<invJ[0][1]<<" "<<invJ[1][0]<<"
"<<invJ[1][1]<<endl;
detJp=J[0][0]*J[1][1]-J[0][1]*J[1][0];
//cout<<detJp<<endl;
```

```
      pfxp[mx]=(invJ[0][0]*kc1[mx]+invJ[0][1]*at1[mx]);

      pfyp[mx]=(invJ[1][0]*kc1[mx]+invJ[1][1]*at1[mx]);

      pfxp[n]=(invJ[0][0]*kc1[n]+invJ[0][1]*at1[n]);

      pfyp[n]=(invJ[1][0]*kc1[n]+invJ[1][1]*at1[n]);
anelement.L11[mx][n]=anelement.L11[mx][n]+pfxp[mx]*pfxp[n]*detJp*alfa*Mu
*Re;
anelement.L22[mx][n]=anelement.L22[mx][n]+pfyp[mx]*pfyp[n]*detJp*alfa*Mu
*Re;
anelement.L21[mx][n]=anelement.L21[mx][n]+pfyp[mx]*pfxp[n]*detJp*alfa*Mu
*Re;
anelement.L12[mx][n]=anelement.L12[mx][n]+pfxp[mx]*pfyp[n]*detJp*alfa*Mu
*Re;
                              at=cp;
                        }
                        kc=cp;
                  }
            }
      }

      for (int bb=0; bb<9; bb++)
      {
            //cout<<endl;
            for (int aa=0; aa<9;aa++)

            {
                  if(fabs(anelement.L21[bb][aa])<10e3)
anelement.L21[bb][aa]=0;
                  if(fabs(anelement.L12[bb][aa])<10e3)
anelement.L12[bb][aa]=0;
                  // cout<<anelement.L12[bb][aa]<<" ";
            }
      }
      anelement.cal_AE12( );
      anelement.cal_AE2( );
      anelement. cal_MF( );
      anelement.cal_global2(nf2);
       }
       //for(int aa=0; aa<70; aa++)
       //{cout<<anelement.GF2[aa]<<endl;}
       //for (int bb=0; bb<70; bb++)
       //{ cout<<endl;
       // for (int aa=0; aa<HBW2;aa++)
         //{
         //    cout<<anelement.Global2[bb][aa]<<" ";}
         //}
boundary2(anelement.Global2, anelement.GF2, nx, ny,
nodenumber,HBW2,NDF2);
         //for(int aa=0; aa<70; aa++)
         // {cout<<anelement.GF2[aa]<<endl;}
         solve2(anelement.Global2, anelement.GF2, nodenumber,HBW2);
         //for (int bb=0; bb<70; bb++)
         // { cout<<endl;
         //  for (int aa=0; aa<HBW2;aa++)
```

```
         //{
         //    cout<<anelement.Global2[bb][aa]<<" ";}
         //}
         for ( lmn=0; lmn<nodenumber; lmn++)
         {
               Ux[lmn]=anelement.GF2[2*lmn];
               Vy[lmn]=anelement.GF2[2*lmn+1];
         }
         ofstream myfile2;
         myfile2.open("velocity.dat", ios::app);
         for(int i=0; i<nodenumber;i++)
         {
               myfile2<<i<<" "<<Ux[i]<<" "<<Vy[i]<<endl;}
         myfile2.close();
       }
       return 0;
}
void calnode(int elenode[9], int x,int y,int ty)
{
      int nyy=2*ty-1;
      elenode[0]=2*nyy*(x-1)+2*(y-1)+1;
      elenode[1]=2*nyy*x+2*(y-1)+1;
      elenode[2]=elenode[1]+2;
      elenode[3]=elenode[0]+2;
      elenode[4]=elenode[0]+1;
      elenode[5]=nyy*(2*x-1)+(2*y-1);
      elenode[6]=elenode[1]+1;
      elenode[7]=elenode[5]+2;
      elenode[8]=elenode[5]+1;
}
void cal_coorx(double x[9], double fxx[2][11], int ii, int jj, int nxx)
{
      x[0]=(fxx[0][jj-2]-fxx[1][jj-2])/(2*nxx-2)*(ii-2)+fxx[1][jj-2];
      x[1]=(fxx[0][jj-2]-fxx[1][jj-2])/(2*nxx-2)*(ii)+fxx[1][jj-2];
              x[2]=(fxx[0][jj]-fxx[1][jj])/(2*nxx-2)*(ii)+fxx[1][jj];
              x[3]=(fxx[0][jj]-fxx[1][jj])/(2*nxx-2)*(ii-2)+fxx[1][jj];
              x[4]=(x[0]+x[3])/2;
              x[5]=(x[0]+x[1])/2;
              x[6]=(x[1]+x[2])/2;
              x[7]=(x[2]+x[3])/2;
              x[8]=(x[0]+x[3]+x[1]+x[2])/4;
}
void cal_coory(double y[9], double fyy[2][81], int i2, int j2, int nyy)
{
      y[0]=(fyy[0][i2-2]-fyy[1][i2-2])/(2*nyy-2)*(j2-2)+fyy[1][i2-2];
              y[1]=(fyy[0][i2]-fyy[1][i2])/(2*nyy-2)*(j2-2)+fyy[1][i2];
              y[2]=(fyy[0][i2]-fyy[1][i2])/(2*nyy-2)*(j2)+fyy[1][i2];
      y[3]=(fyy[0][i2-2]-fyy[1][i2-2])/(2*nyy-2)*(j2)+fyy[1][i2-2];
              y[4]=(y[0]+y[3])/2;
              y[5]=(y[0]+y[1])/2;
              y[6]=(y[1]+y[2])/2;
              y[7]=(y[2]+y[3])/2;
              y[8]=(y[0]+y[3]+y[1]+y[2])/4;
}
void one_element :: define_0()
{
      for (int i1=0; i1<9; i1++)
```

```
        {
                psai_gauss [i1]=0;
                u_gauss [i1]=0;
                v_gauss [i1]=0;
                dxu_gauss [i1]=0;
                dyu_gauss [i1]=0;
                dxv_gauss [i1]=0;
                dyv_gauss [i1]=0;
                dxxu_gauss [i1]=0;
                dyyu_gauss [i1]=0;
                dxxv_gauss [i1]=0;
                dyyv_gauss [i1]=0;
                dxyu_gauss [i1]=0;
                dxyv_gauss [i1]=0;
                dxpsai_gauss [i1]=0;
                dypsai_gauss [i1]=0;
                Dc [i1]=0;
                Ds [i1]=0;
                gamadot [i1]=0;
                dx_gamadot [i1]=0;
                dy_gamadot [i1]=0;
        }
}
void one_element :: define_matrix0()
{
        for (int i1=0; i1<9; i1++)
            {
                for(int i2=0; i2<9; i2++)
                {
                        AE1[i1][i2]=0;
                        S33[i1][i2]=0;
                        M33[i1][i2]=0;
                }
                F3[i1]=0;
                F33[i1]=0;
        }
}
void cal_fai(double fai[9], double att, double kcc)
{
        fai[0]=(kcc*kcc-kcc)*(att*att-att)/4;
        fai[1]=(kcc*kcc+kcc)*(att*att-att)/4;
        fai[2]=(kcc*kcc+kcc)*(att*att+att)/4;
        fai[3]=(kcc*kcc-kcc)*(att*att+att)/4;
        fai[4]=(kcc*kcc-kcc)*(1-att*att)/2;
        fai[5]=(1-kcc*kcc)*(att*att-att)/2;
        fai[6]=(kcc*kcc+kcc)*(1-att*att)/2;
        fai[7]=(1-kcc*kcc)*(att*att+att)/2;
        fai[8]=(1-kcc*kcc)*(1-att*att);
}
void cal_kc1(double d1kc[9], double att, double kcc)
{
        d1kc[0]=(2*kcc-1)*(att*att-att)/4;
        d1kc[1]=(2*kcc+1)*(att*att-att)/4;
        d1kc[2]=(2*kcc+1)*(att*att+att)/4;
        d1kc[3]=(2*kcc-1)*(att*att+att)/4;
        d1kc[4]=(2*kcc-1)*(1-att*att)/2;
        d1kc[5]=-kcc*(att*att-att);
```

```
        d1kc[6]=(2*kcc+1)*(1-att*att)/2;
        d1kc[7]=-kcc*(att*att+att);
        d1kc[8]=-2*kcc*(1-att*att);
}
void cal_kc2(double d2kc[9], double att, double kcc)
{
        d2kc[0]=(att*att-att)/2;
        d2kc[1]=(att*att-att)/2;
        d2kc[2]=(att*att+att)/2;
        d2kc[3]=(att*att+att)/2;
        d2kc[4]=(1-att*att);
        d2kc[5]=-(att*att-att);
        d2kc[6]=(1-att*att);
        d2kc[7]=-(att*att+att);
        d2kc[8]=-2*(1-att*att);
}
void cal_at1(double d1at[9], double att, double kcc)
{
        d1at[0]=(2*att-1)*(kcc*kcc-kcc)/4;
        d1at[1]=(2*att-1)*(kcc*kcc+kcc)/4;
        d1at[2]=(2*att+1)*(kcc*kcc+kcc)/4;
        d1at[3]=(2*att+1)*(kcc*kcc-kcc)/4;
        d1at[4]=-att*(kcc*kcc-kcc);
        d1at[5]=(2*att-1)*(1-kcc*kcc)/2;
        d1at[6]=-att*(kcc*kcc+kcc);
        d1at[7]=(2*att+1)*(1-kcc*kcc)/2;
        d1at[8]=-2*att*(1-kcc*kcc);
        //for (int i=0; i<9; i++)
        //{ cout<<d1at[i]<<endl;}
}
void cal_at2(double d2at[9], double att, double kcc)
{
        d2at[0]=(kcc*kcc-kcc)/2;
        d2at[1]=(kcc*kcc+kcc)/2;
        d2at[2]=(kcc*kcc+kcc)/2;
        d2at[3]=(kcc*kcc-kcc)/2;
        d2at[4]=-(kcc*kcc-kcc);
        d2at[5]=(1-kcc*kcc);
        d2at[6]=-(kcc*kcc+kcc);
        d2at[7]=(1-kcc*kcc);
        d2at[8]=-2*(1-kcc*kcc);
}
void cal_kcat(double dkcat[9], double att, double kcc)
{
        dkcat[0]=(2*kcc-1)*(2*att-1)/4;
        dkcat[1]=(2*kcc+1)*(2*att-1)/4;
        dkcat[2]=(2*kcc+1)*(2*att+1)/4;
        dkcat[3]=(2*kcc-1)*(2*att+1)/4;
        dkcat[4]=-(2*kcc-1)*att;
        dkcat[5]=-(2*att-1)*kcc;
        dkcat[6]=-(2*kcc+1)*att;
        dkcat[7]=-(2*att+1)*kcc;
        dkcat[8]=4*kcc*att;
}
void cal_J(double array1[2][2], double a1[9], double b1[9], double
c1[9],
           double d1[9])
```

```
{
      for (int i=0; i<9; i++)
      {
            array1[0][0]=array1[0][0]+a1[i]*c1[i];
            array1[1][1]=array1[1][1]+b1[i]*d1[i];
            array1[0][1]=array1[0][1]+a1[i]*d1[i];
            array1[1][0]=array1[1][0]+b1[i]*c1[i];}
      for (int a=0; a<2;a++)
      {
            for(int b=0; b<2; b++)
            {
                  if(fabs(array1[a][b])<10e-5) array1[a][b]=0;}
      }
}
void cal_invJ(double inJ[2][2], double array2[2][2])
{
      double delJ;
      delJ=array2[0][0]*array2[1][1]-array2[0][1]*array2[1][0];
      inJ[0][0]=array2[1][1]/delJ;
      inJ[1][1]=array2[0][0]/delJ;
      inJ[0][1]=-array2[0][1]/delJ;
      inJ[1][0]=-array2[1][0]/delJ;
      for (int a=0; a<2;a++)
      {
            for(int b=0; b<2; b++)
            {
                  if(fabs(inJ[a][b])<10e-5) inJ[a][b]=0;}
      }
}
void cal_J1(double array2[3][3], double array1[2][2])
{
      array2[0][0]=array1[0][0]*array1[0][0];
      array2[0][1]=array1[0][1]*array1[0][1];
      array2[0][2]=2*array1[0][0]*array1[0][1];
      array2[1][0]=array1[1][0]*array1[1][0];
      array2[1][1]=array1[1][1]*array1[1][1];
      array2[1][2]=2*array1[1][0]*array1[1][1];
      array2[2][0]=array1[0][0]*array1[1][0];
      array2[2][1]=array1[0][1]*array1[1][1];
      array2[2][2]=array1[1][0]*array1[0][1]+array1[0][0]*array1[1][1];
}
void cal_invJ1(double array2[3][3], double array1[3][3])
{
      double c1, c2, c3, delc;
      c1=array1[1][1]*array1[2][2]-array1[1][2]*array1[2][1];
      c2=array1[1][2]*array1[2][0]-array1[1][0]*array1[2][2];
      c3=array1[1][0]*array1[2][1]-array1[1][1]*array1[2][0];
      delc=array1[0][0]*c1+array1[0][1]*c2+array1[0][2]*c3;
array2[0][0]= array1[1][1]*array1[2][2]-array1[2][1]*array1[1][2]/delc;
array2[0][1]= -(array1[0][1]*array1[2][2]-
array1[0][2]*array1[2][1])/delc;
array2[0][2]= array1[0][1]*array1[1][2]-array1[0][2]*array1[1][1]/delc;
array2[1][0]= -(array1[1][0]*array1[2][2]-
array1[1][2]*array1[2][0])/delc;
array2[1][1]= array1[0][0]*array1[2][2]-array1[2][0]*array1[0][2]/delc;
array2[1][2]= -(array1[0][0]*array1[1][2]-
array1[0][2]*array1[1][0])/delc;
```

```
array2[2][0]= array1[1][0]*array1[2][1]-array1[2][0]*array1[1][1]/delc;
array2[2][1]= -(array1[0][0]*array1[2][1]-
array1[0][1]*array1[2][0])/delc;
array2[2][2]= array1[0][0]*array1[1][1]-array1[1][0]*array1[0][1]/delc;
}

void cal_J2(double array1[3][2], double a1[9], double b1[9],  double
ab[9],            double c1[9], double d1[9])
{
    for(int i=0; i<9; i++)
    {
            array1[0][0]=array1[0][0]+a1[i]*c1[i];
            array1[0][1]=array1[0][1]+a1[i]*d1[i];
            array1[1][0]=array1[1][0]+b1[i]*c1[i];
            array1[1][1]=array1[1][1]+b1[i]*d1[i];
            array1[2][0]=array1[2][0]+ab[i]*c1[i];
            array1[2][1]=array1[2][1]+ab[i]*d1[i];
      }
}
void one_element::cal_F()
{
      int i,j;
      for( i=0; i<9; i++)
      {
            F3[i]=F33[i];
            for(j=0; j<9; j++)
            {
                  F3[i]=F3[i]+M33[i][j]*psai[j] ;
            }
            //cout<<F3[i]<<endl;
      }
}
void one_element:: cal_AE1()
{
      int i, j;
      for (i=0; i<9; i++)
      {
            //cout<<endl;
            for (j=0; j<9; j++)
            {
                  AE1[i][j]=M33[i][j]+Deltat*S33[i][j];
                  //cout<<AE1[i][j]<<" ";
            }
      }
}
void one_element::cal_global1(int nf1)
{
      for (int i=1; i<=9; i++)
      {
            int NR=(ndnumber[i-1])*nf1;
            for (int ii=1; ii<=nf1; ii++)
            {
                  NR=NR+1;
                  int Le=(i-1)*nf1+ii;
                  GF1[NR-1]=GF1[NR-1]+F3[Le-1];
                  for (int j=1; j<=9; j++)
                  {
```

```
                                int NCL=(ndnumber[j-1])*nf1;
                                for (int jj=1; jj<=nf1; jj++)
                                {
                                        int Me=(j-1)*nf1+jj;
                                        int NC=NCL-NR+jj+1;
                                        if (NC>0)
                                        {
                Global1[NR-1][NC-1]=Global1[NR-1][NC-1]+AE1[Le-1][Me-1];}
                                }
                        }
                }
        }
}
void boundary1(double ag[711][21], double bf[711], int cny, int
dnodenumber, int ehbw1, int fnf1)
{
        int ISPV[10];
        double VSPV[10];
        int NEQ1=dnodenumber;
        int NSPV=2*cny-1;
        for (int b_num=1; b_num<=NSPV; b_num++)
        {
                ISPV[b_num]=(b_num);
                VSPV[b_num]=0.30;
        }
        for (int NB=1; NB<=NSPV; NB++)
        {
                int IE, It, I;
                IE=(ISPV[NB]-1)*fnf1+1;
                It=ehbw1-1;
                I=IE-ehbw1;
                for ( int ii=1; ii<=It; ii++)
                {
                        I=I+1;
                        if(I>=1)
                        {
                                int j=IE-I+1;
                                bf[I-1]=bf[I-1]-ag[I-1][j-1]*VSPV[NB];
                                ag[I-1][j-1]=0;}
                }
                ag[IE-1][0]=1.0;
                bf[IE-1]=VSPV[NB];
                I=IE;
                for (int ii1=2;ii1<=ehbw1; ii1++)
                {
                        I=I+1;
                        if (I<=NEQ1)
                        {
                                bf[I-1]=bf[I-1]-ag[IE-1][ii1-1]*VSPV[NB];
                                ag[IE-1][ii1-1]=0.0;
                        }
                }
        }
}
void solve1(double ag[711][21], double bf[711], int cnodenumber, int
dhbw1)
{
```

```
    int MEQNS=cnodenumber-1;
    for (int NPIV=1; NPIV<=MEQNS; NPIV++)
    {
        int NPIVOT=NPIV+1;
        int LSTSUB=NPIV+dhbw1-1;
        if(LSTSUB>cnodenumber)
        {
            LSTSUB=cnodenumber;}
        for (int Nrow=NPIVOT; Nrow<=LSTSUB; Nrow++)
        {
            int Ncol=Nrow-NPIV+1;
            double Factor=ag[NPIV-1][Ncol-1]/ag[NPIV-1][0];
            for ( Ncol=Nrow;Ncol<=LSTSUB; Ncol++)
            {
                int Icol=Ncol-Nrow+1;
                int Jcol=Ncol-NPIV+1;
        ag[Nrow-1][Icol-1]=ag[Nrow-1][Icol-1]-Factor*ag[NPIV-
1][Jcol-1];
            }
            bf[Nrow-1]=bf[Nrow-1]-Factor*bf[NPIV-1];
        }
    }
    //back substitution
    for (int IJK=2;IJK<=cnodenumber; IJK++)
    {
        int  NPIV=cnodenumber-IJK+2;
        bf[NPIV-1]=bf[NPIV-1]/ag[NPIV-1][0];
        int LSTSUB=NPIV-dhbw1+1;
        if(LSTSUB<1)
        {
            LSTSUB=1;}
        int NPIVOT=NPIV-1;
        for (int JKI=LSTSUB; JKI<=NPIVOT; JKI++)
        {
            int Nrow=NPIVOT-JKI+LSTSUB;
            int Ncol=NPIV-Nrow+1;
            double Factor=ag[Nrow-1][Ncol-1];
            bf[Nrow-1]=bf[Nrow-1]-Factor*bf[NPIV-1];
        }
    }
    bf[0]=bf[0]/ag[0][0];
}
void one_element :: define_matrix02()
{
    for (int i1=0; i1<9; i1++)
    {
        for(int i2=0; i2<9; i2++)
        {
            AE2[i1][i2]=0;
            AE12[i1][i2]=0;
            AE2[i1+9][i2+9]=0;
            AE12[i1+9][i2+9]=0;
            S11[i1][i2]=0;
            S22[i1][i2]=0;
            M11[i1][i2]=0;
            M22[i1][i2]=0;
        }
```

```
            F1[i1]=0;
            F2[i1]=0;
            F11[i1]=0;
            F22[i1]=0;
            F[i1]=0;
            F[i1+9]=0;
        }
}
void one_element :: define_matrix03()
{
        for (int i1=0; i1<9; i1++)
        {
            for(int i2=0; i2<9; i2++)
            {
                L11[i1][i2]=0;
                L12[i1][i2]=0;
                L21[i1][i2]=0;
                L22[i1][i2]=0;
            }
        }
}
void one_element :: define_02()
{
        for (int i1=0; i1<9; i1++)
        {
            psai_gauss [i1]=0;
            u_gauss [i1]=0;
            v_gauss [i1]=0;
            dxu_gauss [i1]=0;
            dyu_gauss [i1]=0;
            dxv_gauss [i1]=0;
            dyv_gauss [i1]=0;
            dxpsai_gauss [i1]=0;
            dypsai_gauss [i1]=0;
        }
}
void one_element::cal_F1() {
        int i,j;
        for( i=0; i<9; i++)
        {
            F1[i]=F11[i];
            for(j=0; j<9; j++)
            {
                F1[i]=F1[i]+M11[i][j]*x_velocity[j] ;
            }
            //cout<<F1[i]<<endl;
        }
}
void one_element::cal_F2()
{
        int i,j;
        for( i=0; i<9; i++)
        {
            F2[i]=F22[i];
            for(j=0; j<9; j++)
            {
                F2[i]=F2[i]+M22[i][j]*y_velocity[j] ;
```

```
                }
                //cout<<F2[i]<<endl;
        }
}
void one_element :: cal_AE12( )
{
        for (int m=0; m<9; m++)
        {
                for (int n=0; n<9; n++)
                {
                        AE12[m][n]=M11[m][n]+Deltat*(S11[m][n]+L11[m][n]);
                        AE12[m][n+9]=Deltat*L12[m][n];
                        AE12[m+9][n]=Deltat*L21[m][n];
                        AE12[m+9][n+9]=M22[m][n]+Deltat*(S22[m][n]+L22[m][n]);
                }
        }
}
void one_element :: cal_AE2( )
{
        int y1, y2;
        for (int m=1; m<=18; m++)
        {
                for (int n=1; n<=18; n++)
                {
                        int x1=int((n-1)/9)+1;
                        if ( n==9 || n==18)    y1=9;
                        else y1=n%9;
                        int x2=int(( m-1)/9)+1;
                        if (m==9 || m==18)
                                y2=9;
                        else y2=m%9;
                        int nn=(y1-1)*2+x1;
                        int mm=(y2-1)*2+x2;
                        AE2[mm-1][nn-1]=AE12[m-1][n-1];
                }
        }
}
void one_element :: cal_MF( )
{
        for (int m=0; m<9; m++)
        {
                F[m*2]=F1[m];
                F[m*2+1]=F2[m];
        }
}
void one_element::cal_global2(int nf2)
{
        for (int i=1; i<=9; i++)
        {
                int NR=(ndnumber[i-1])*nf2;
                for (int ii=1; ii<=nf2; ii++)
                {
                        NR=NR+1;
                        int Le=(i-1)*nf2+ii;
                        GF2[NR-1]=GF2[NR-1]+F[Le-1];
                        for (int j=1; j<=9; j++)
                        {
```

```
                        int NCL=(ndnumber[j-1])*nf2;
                        for (int jj=1; jj<=nf2; jj++)
                        {
                               int Me=(j-1)*nf2+jj;
                               int NC=NCL-NR+jj+1;
                               if (NC>0)
                               {
          Global2[NR-1][NC-1]=Global2[NR-1][NC-1]+AE2[Le-1][Me-1];}
                        }
                  }
            }
      }
}
void boundary2(double ag[1422][42], double bf[1422],int cnx, int cny,
               int dnodenumber, int ehbw2, int fnf2)
{
      int ISPV[166];
      double VSPV[166][2];
      int NEQ2=2*dnodenumber;
      int NSPV=(2*cnx-1)*2+(2*cny-3);
      for (int b_num=1; b_num<=NSPV; b_num++)

      {
            if (b_num<=(2*cnx-1)*2&&(b_num%2==1))
            {
                  ISPV[b_num]=1+((b_num+1)/2-1)*(2*cny-1);
                  VSPV[b_num][0]=0;
                  VSPV[b_num][1]=0;
            }
            if (b_num<=(2*cnx-1)*2 && (b_num%2==0))
            {
                  ISPV[b_num]=(b_num/2)*(2*cny-1);
                  VSPV[b_num][0]=0;
                  VSPV[b_num][1]=0;
            }
            if (b_num>(2*cnx-1)*2 && b_num<=(2*cnx-1)*2+(2*cny-3))
            {
                  ISPV[b_num]=(b_num-(2*cnx-1)*2)+1;
                  VSPV[b_num][0]=1.0;
                  VSPV[b_num][1]=0;
            }
      }
      for (int NB=1; NB<=NSPV; NB++)
      {
            int IE=(ISPV[NB]-1)*fnf2+1;
            int It=ehbw2-1;
            int I=IE-ehbw2;
            for ( int ii=1; ii<=It; ii++)
            {
                  I=I+1;
                  if(I>=1)
                  {
                        int j=IE-I+1;
                        bf[I-1]=bf[I-1]-ag[I-1][j-1]*VSPV[NB][0];
                        ag[I-1][j-1]=0;}
            }
            ag[IE-1][0]=1.0;
```

```
            bf[IE-1]=VSPV[NB][0];
            I=IE;
            for (int ii1=2;ii1<=ehbw2; ii1++)
            {
                  I=I+1;
                  if (I<=NEQ2)
                  {
                        bf[I-1]=bf[I-1]-ag[IE-1][ii1-1]*VSPV[NB][0];
                        ag[IE-1][ii1-1]=0.0;
                  }
            }
      }
      for (int NB2=1; NB2<=NSPV; NB2++)
      {
            int IE=(ISPV[NB2]-1)*fnf2+2;
            int It=ehbw2-1;
            int I=IE-ehbw2;
            for ( int ii=1; ii<=It; ii++)
            {
                  I=I+1;
                  if(I>=1)
                  {
                        int j=IE-I+1;
                        bf[I-1]=bf[I-1]-ag[I-1][j-1]*VSPV[NB2][1];
                        ag[I-1][j-1]=0;}
            }
            ag[IE-1][0]=1.0;
            bf[IE-1]=VSPV[NB2][1];
            I=IE;
            for (int ii2=2;ii2<=ehbw2; ii2++)
            {
                  I=I+1;
                  if (I<=NEQ2)
                  {
                        bf[I-1]=bf[I-1]-ag[IE-1][ii2-1]*VSPV[NB2][1];
                        ag[IE-1][ii2-1]=0.0;
                  }
            }
      }
}
void solve2(double ag[1422][42], double bf[1422], int cnodenumber, int
dhbw2)
{
      int MEQNS=cnodenumber*2-1;
      int NEQ2=cnodenumber*2;
      for (int NPIV=1; NPIV<=MEQNS; NPIV++)
      {
            int NPIVOT=NPIV+1;
            int LSTSUB=NPIV+dhbw2-1;
            if(LSTSUB>NEQ2)
            {
                  LSTSUB=NEQ2;}
            for (int Nrow=NPIVOT; Nrow<=LSTSUB; Nrow++)
            {
                  int Ncol=Nrow-NPIV+1;
                  double Factor=ag[NPIV-1][Ncol-1]/ag[NPIV-1][0];
                  for ( Ncol=Nrow;Ncol<=LSTSUB; Ncol++)
```

```
            {
                int Icol=Ncol-Nrow+1;
                int Jcol=Ncol-NPIV+1;
    ag[Nrow-1][Icol-1]=ag[Nrow-1][Icol-1]-Factor*ag[NPIV-1][Jcol-1];
            }
            bf[Nrow-1]=bf[Nrow-1]-Factor*bf[NPIV-1];
        }
    }
    //back substitution
    for (int IJK=2;IJK<=NEQ2; IJK++)
    {
        int  NPIV=NEQ2-IJK+2;
        bf[NPIV-1]=bf[NPIV-1]/ag[NPIV-1][0];
        int LSTSUB=NPIV-dhbw2+1;
        if(LSTSUB<1)
        {
            LSTSUB=1;}
        int NPIVOT=NPIV-1;
        for (int JKI=LSTSUB; JKI<=NPIVOT; JKI++)
        {
            int Nrow=NPIVOT-JKI+LSTSUB;
            int Ncol=NPIV-Nrow+1;
            double Factor=ag[Nrow-1][Ncol-1];
            bf[Nrow-1]=bf[Nrow-1]-Factor*bf[NPIV-1];
        }
    }
    bf[0]=bf[0]/ag[0][0];
}
```

REFERENCES

1. Rudinger, G. (1980). *Fundamentals of Gas-Particle Flow.* Elsevier Scientific Publishing Company, Amsterdam.

2. Zhang, K., Acrivos, A. (1994). "Viscous Resuspension in Fully Developed LaminerPipe Flow." *International Journal of Multiphase flow*, Vol. 20, No. 3, 579-591.

3. Sinton, S. W., Chow, A. W. (1991). "NMR Flow Imaging of Fluids and Solid Suspensions Inpoiseuille Flow." *Journal of Rheology*, 35(5), 735-772.

4. Miskin, I., Elliott, L., Ingham, D. B., Hammond, P. S. (1995)." The Viscous Resuspension of Particles in an Inclined Rectangular Fracture." *International Journal of Multiphase Flow*, Vol. 22, No.2, 403-415.

5. Soo, S. L. (1989). *Particulates and Continuum Multiphase Fluid Dynamics.* Hemisphere Publishing Corporation, NY.

5. Acrivos, A. (1990). "Trends in Liquid-Solid Flow Research." *NSF-DOE Workshop on Flow of Particulates and Fluids*, MD.

6. Batchelor, G. K., Green, J. T. (1972). "The Hydrodynamic Interaction of Two Small Freely-Moving Spheres in a Linear Flow Field." *Journal of Fluid Mechanics*, 56, 375-401.

7. Gadala-Maria, F., Acrivos, A. (1980). "Shear-Induced Structure in a Concentrated Suspension of Solid Spheres." *Journal of Rheology*, 24, 799-814.

8. Leighton, D., Acrivos, A. (1987). "The Shear-Induced Migration of Particles in Concentrated Suspensions." *Journal of Fluid Mechanics*, vol. 181, 415-430.

9. Leighton, D., Acrivos, A. (1986). "Measurement of Shear-Induced Self-Diffusion in Concentrated Suspensions of Spheres." *Journal of Fluid Mechanics*, Vol. 177, 109-131.

10. Leighton, D., Acrivos, A. (1986). "Viscous Resuspension." *Chemical Engineering Science*. 41,1377-1384.

11. Graham, A. L., Altobelli, S. A., Fukushima, E., Mondy, L. A., Stephens, T. S. (1991). "Note: NMR Imaging of Shear-Induced Diffusion and Structure in Concentrated Suspensions Undergoing Couette Flow." *Journal of Rheology*, 35(1), 191-201.

12. Chapman, B. K., Leighton, D. T. JR (1991). "Dynamic Viscous Resuspension." *International Journal of Multiphase Flow*, Vol. 17, No. 4, 469-483.

13. Schaflinger, U., Acrivos, A., Stibi, H. (1995) "An Experimental Study of Viscous Resuspension in a Pressure-Driven Plane Channel Flow." *International Journal of Multiphase Flow*, Vol. 21, No. 4, 693-704.

14. Acrivos, A., Mauri, R., Fan, X. (1993). "Shear-Induced Resuspension in a Couette Device." *International Journal of Multiphase Flow*, Vol. 19, 797-802.

15. Phillips, R. J., Armstrong, R. C., Brown, R. A., Graham, A. L., Abbott, J. R. (1991). "A Constitutive Equation for Concentrated Suspensions That Accounts for Shear-Induced Particle Migration." *Physics of Fluids*, A4(1), 30-40.

16. Altobelli, S. A., Givler, R. C., Fukushima, E. (1991). "Velocity and Concentration Measurements of Suspension by Nuclear Magnetic Resonance." *Journal of Rheology*, 35(5), 721-734.

17. Abbott, J. R., Graham, A. L., Tetlow, N., Altobelli, S. A., Fukushima, E., Mondy, L. A., Stephens, T. S. (1991). "Experimental Observations of Particle Migreation In Concentrated Suspension: Couette Flow." *Journal of Rheology*, 35(5), 773-795.

18. Chow, A. W., Sinton, S. W., Iwamiya, J. H. (1994). "Shear-Induced Particle Migration in Couette and Parallel-Plate Viscometer: NMR Imaging and Stress Measurements." *Physics of Fluids*, Vol. 6, No. 8, 2561-2578.

19. Hampton, R. E., Mammoli, A. A., Graham, A. L., Tetlow, N, Altobelli, S. A. (1997). "Migration of Particle Undergoing Pressure-Driven Flow in a Circular Conduit." *Journal of Rheology*, 41(3), 621-640.

20. Mondy, L. A., Brenner, H., Altobelli, S. A., Abbott, J. R., Graham, A. L. (1993). "Shear-Induced Particle Migration in Suspensions of Rods." *Journal of Rheology*, 38(2), 444-452.

21. Koh, C. J., Hookham, P., Leal, L. G. (1993). "An Experimental Investigation of Concentrated Suspension Flows in a Rectangular Channel." *Journal of Fluid Mechanics*, Vol. 266, 1-32.

22. Jenkins, J. T., McTigue, D. F. (1990) "Transport Process in Concentrated Suspensions: The Role of Particle Fluctuations." *Two Phase Flows and Waves*, Springer.

23. Nott, P. R., Brady, J. F. (1994). "Pressure-Driven Flow of Suspensions: Simulation and Theory." *Journal of Fluid Mechanics*, Vol. 275, 157-199.

24. Brady, J. F., Bossis, G. (1988). "Stokesian Dynamics." *Annual Review of Fluid Mechanics*, 20, 111-157.

25. Miskin, I., Elliott, L., Ingham, D. B., Hammond, P. S. (1995). "The Viscous Resuspension of Particles in an Inclined Rectangular Fracture." *International Journal of Multiphase Flow*, Vol. 22, 403-415.

26. Miskin, I., Elliott, L., Ingham, D. B., Hammond, P. S. (1995). "Steady Suspension Flows into Two Dimensional Horizontal and Inclined Channels." *International Journal of Multiphase Flow*, Vol. 22, 1223-1246.

27. Fang, Z., Phan-Thien, N. (1995). "Numerical Simulation of Particle Migration in Concentrated Suspensions by a Finite Volume Method." *Journal of Non-Newtonian Fluid Mechanics*, 58, 67-81.

28. Schaflinger, U., Acrovis, A., Zhang, K. (1993). "Viscous Resuspension of a Sediment Within a Laminar and Stratified Flows." *International Journal of Multiphase Flow*, Vol. 16, No. 3, 567-578.

29. Nir, A., Acrovis, A. (1989). "Sedimentation and Sediment Flow on Inclined Surfaces." *Journal of Fluid Mechanics*, Vol. 212, 130-153.

30. Pelekasis, N. A., Acrovis, A. (1995). "Forced Convection and Sedimentation Past a Flat Plate." *Journal of Fluid Mechanics*, Vol. 294, 301-321.

31. Phan-thien, N., Fang, Z. (1996). "Entrance Length and Pulsatile Flows of a Model Concentrated Suspension." *Journal of Rheology*, 40 (4), 521-529.

32. Phan-Thien, N., Graham, A. L., Altobelli, S. A., Abbott, J. R., Mondy, L. A. (1995). "Hydrodynamic Particle Migration in a Concentrated Suspension Undergoing Flow Between Rotating Eccentric Cylinders." *Ind. Eng. Chem. Res.* , 34, 3187-3194.

33. Krishnan, G. P., Leighton, D. T. Jr. (1995). "Dynamic Viscous Resuspension of Bidisperse Suspensions—I. Effective Diffusivity." *International Journal of Multiphase Flow*, Vol. 21, 721-732.

34. Krishnan, G. P., Beimfohr, S., Leighton, D. T. (1996). "Shear-Induced Radial Segregation in Bidisperse Suspensions." *Journal of Fluid mechanics*, Vol. 321, 371-393.

35. Subia, S. R., Ingber, M. S., Mondy, L. A., Altobelli, A. A., Graham, A. L. (1998). "Modelingof Concentrated Suspensions Using a Continuum Constitutive Equation." *Journal of Fluid Mechanics*, Vol. 373, 193-219.

36. Averbakh, A., Shauly, A., Nir, A., Semiat, R. (1996). "Slow Viscous Flows of Highly Concentrated Suspensions-Part I: Laser-Doppler Velocimetry in Rectangular Ducts." *International Journal of Multiphase Flow*, Vol. 23, 409-424.

37. Reddy, J. N. (1993) *An Introduction to the Finite Element Method.* McGraw-Hill, Inc, NJ.

38. Cuverlier, C., Segal, A. (1986). *Finite Element Methods and Navier-Stokes Equations.* D. Reidel publishing Company, NY.

39. Hughes, T. J. R., Liu, W. K., Brooks, A. (1979). "Finite Element Analysis of Incompressible Viscous Flow by the Penalty Function Formulation". *Journal of Computing Physics*, 30, 1-60.

40. Chen, R-Y (1973), "Flow in the Entrance Region at Low Reynolds Number". *Journal of Fluid Engineering, March*, 1973, 153-158.

41. Ingham, D. B. (1980) " Deposition of Charged particle Near the Entrance of a Cylindrical Tube". *Journal of Aerosol Science*, 11, 47-52