MULTIPHASE HYDRODYNAMICS OF GAS-SOLIDS FLOW

BY

M. SYAMLAL

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering
in the School of Advanced Studies of
Illinois Institute of Technology

Approved D. Gidaspow
Adviser

Chicago, Illinois
December, 1985

MULTIPHASE HYDRODYNAMICS OF GAS-SOLIDS FLOW

BY

M. SYAMLAL

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering
in the School of Advanced Studies of
Illinois Institute of Technology

Approved D. Gidaspow
Adviser

Chicago, Illinois
December, 1985

ACKNOWLEDGMENT

I am indebted to a great many people who helped me at various stages of the thesis preparation. I would like to express my gratitude to Dr. Dimitri Gidaspow for introducing me to the fascinating field of computational fluid dynamics and for providing valuable guidance in technical matters which lead to the successful completion of this study. Thanks are also due to all my friends at IIT for their help and for the many stimulating dicsussions. The cooperation of Ms. Elaine Mann and Ms. Lillian Courtney who skillfully typed most of this thesis is also being acknowledged.

I am grateful to my parents whose encouragement, and good wishes guided me through difficult times and Girija, my wife, for her love patience, and understanding.

Financial support for this study was provided by the National Science Foundation grant, "Hydrodynamics of Fluidization", CPE 8209290; U.S. Department of Energy grant, "Dilute, Dense Phase and Maximum Solid-Gas Transport", DOE grant No. DE-FG 22-84 PC 70057; and private industry.

M.S.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT	iii
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF SYMBOLS	x
ABSTRACT	xiii
CHAPTER	
I. INTRODUCTION	1.1
II. MULTIPHASE HYDRODYNAMICS	2.1
The Governing Equations	2.1
Gas-Particle Interaction	2.4
Particle-Particle Interaction	2.7
The Solids Stress Term	2.15
The Intial and Boundary Conditions	2.17
III. MIXING OF PARTICLES IN A FLUIDIZED BED	3.1
IV. HEAT TRANSFER FROM A FLUIDIZED BED	4.1
The Energy Equations	4.3
Boundary and Initial Conditions	4.4
Interphase Heat Transfer	4.6
Particulate Phase Heat Transfer	4.9
Numerical Solution	4.14
Results and Discussions	4.17
V. DILUTE PHASE TRANSPORT	5.1
Radial Velocity Profiles	5.1
Segregation in a Pneumatic Conveyor	5.18
VI. CRITICAL MULTIPHASE FLOW	6.1
Steady Gas-Solids Flow	6.2
Discharge From a Hopper	6.7
Critical Flow From a Fluidized Bed	6.7
VII. SUMMARY AND RECOMMENDATIONS	7.1

Page

APPENDIX

A. A MANUAL FOR THE MULTIFIX PROGRAM 8.1

B. SOME PRELIMINARY RESULTS 9.1

BIBLIOGRAPHY 10.1

LIST OF TABLES

Table		Page
2.1.	Comparison of Minimum Fluidization Velocities of Multiparticle Systems	2.8
3.1.	Simulation Conditions for a Two Particle System	3.2
3.2.	Comparison of Fractional Bed Expansion Data	3.10
4.1.	Interphase Heat Transfer Correlations	4.8
4.2.	Conditions for the Heat Transfer Simulation	4.17
5.1.	Conditions for the Pneumatic Conveyor Simulation	5.2
5.2.	Comparison of Two Pneumatic Conveyor Simulations	5.16

LIST OF FIGURES

Figure		Page
1.1.	The Evolution and Propagation of a Bubble in a Bed of 800 μm Particles (jet velocity = 5 m/s)	1.3
1.2.	Solids Velocity Vector Plots for The Same Simulation as Shown in Figure 1.1	1.4
2.1.	Pressure Drop Versus Gas Velocity for a Bed of Two Particle Size	2.6
2.2.	Definition Sketch for a Two Particle System	2.9
2.3.	Definition Sketch for a Collision Pair	2.12
3.1.	Geometry of the System for the Mixing Experiment	3.3
3.2.	Volume Fraction Plots for the Ballotine Particles at Various Times	3.5
3.3.	Volume Fraction Plots for the Lighten Particles at Various Times	3.7
3.4.	Comparison of the Drift Layer as Predicted by the Theory to Experimentally Observed Interface When the Bed is Collapsed at the end of 0.5s	3.8
3.5	Experimentally Observed Interfaces When the Bed is in Operation	3.9
3.6.	Gas Velocity Vector Plots at Various Times	3.12
3.7.	Solids Velocity Vector Plots of the Dense Bottom Layer	3.13
3.8.	Solids Velocity Vector Plots for the Top Layer of Light Particles	3.14
4.1.	Typical Data for Numerical Computations	4.5
4.2.	Fluid-Particle Heat Transfer	4.7
4.3.	Effective Bed Conductivity as a Function of Porosity	4.11
4.4.	Zehna and Schlunder Model for Effective Radial Conductivity of Solids in a Packed Bed	4.13
4.5.	Convergence as the Number of Subcells is Increased	4.16
4.6.	Average Bed-to-Wall Heat Transfer Coefficients for Different Gas Flow Rates	4.18

4.7. Comparison of Average Heat Transfer Coefficients With Experimental Data. 4.20

4.8. Gas Temperature Profiles Near the Wall 4.21

4.9. Local Heat Transfer Coefficients at Two Locations in the Bed 4.23

4.10. ε, V_g, and V_s at 0.11m Above the Distributor ($U-U_{mf}$ = 0.16 m/s) 4.24

4.11. Contributions to Heat Transfer Due to Gas Phase and Particulate Phase Conduction 4.25

5.1. Experimental Axial Velocity Profiles 5.3

5.2. Axial Gas Velocity Profiles for μ_s = 1 Poise 5.4

5.3. Axial Solids Velocity Profiles for μ_s = 1 Poise . . . 5.5

5.4. Solids Volume Fraction Profiles for μ_s = 1 Poise . . 5.7

5.5. Axial Gas Velocity Profiles for μ_s = 10 Poise 5.8

5.6. Axial Solids Velocity Profiles for μ_s = 10 Poise . . . 5.9

5.7. Solids Volume Fraction Profiles for μ_s= 10 Poise . . . 5.10

5.8. Solids Volume Fraction Profiles at a Height of 4m Above the Entrance for μ_s = 10 Poise 5.11

5.9. Gas Velocity Profiles for a Higher Solids Loading. . . 5.12

5.10. Solids Velocity Profiles for a Higher Solids Loading . 5.13

5.11. Solids Volume Fraction Profiles for a Higher Solids Loading 5.14

5.12. Development of the Axial Gas Velocity at the Center . . 5.15

5.13. Development of the Axial Solids Velocity at the Center . 5.17

5.14. Pneumatic Conveying of a Binary Mixture 5.19

5.15. Solids Velocity Along the Length of a Pneumatic Conveyor 5.21

5.16. Segregation in a Pneumatic Conveyor as a Function of the Gas Velocity 5.22

5.17. Comparison of the Experimental and the Computed Solids Velocities in a Pneumatic Conveyor 5.23

6.1. Computational Region for the Critical Flow Calculations 6.9

6.2. Transient Behavior of the Local Variables at the Orifice 6.10

8.1. The Computational Mesh 8.6

8.2. The Staggered Computational Mesh for Momentum Equations 8.7

8.3. The Iteration Loop 8.13

8.4. The Secant Methods for Pressure Correction 8.14

8.5. Program Flow Chart 8.19

9.1. Initial Conditions for the Fines Injection Simulation 9.3

9.2. Distribution of Fines Injected Through a Central Jet into a Bed of Coarse Particles 9.4

9.3. Initial Conditions for the Segregation Simulation. . . . 9.5

9.4. Volume Fractions of the Two Particulate Phases at Various Locations in the Bed 9.6

9.5. Initial Conditions for the Mixing Simulation 9.9

9.6. Solids Volume Fraction Contours at the End of Two Seconds in a Segregated Bed 9.10

LIST OF SYMBOLS

Symbol	Term
C_{Dk}	Gas particulate phase k drag coefficient
C_g	Speed of sound in gas $\sqrt{\left(\frac{\alpha P}{\alpha \rho}\right)_T}$ (cm/s)
$D^j{}_i$	Residue of gas continuity equation (24) (g/cm^3)
d_k	Characteristic diameter of particulate phase k (cm)
e	Coefficient of restitution
f	Function deined by equation 2.7
g	Accleration due to gravity (cm/s^2)
g1	Constant in equation 2.39 (dynes.cm^2)
g2	Constant in equation 2.39
G_k	Solid stress modulus (gm/cm s^2)
h_i, h_{av}	Wall to bed heat transfer coefficients, W/m^2°K
h_1	
h_v	Interphase heat transfer coefficient, W/m^3 °K
I_g	Specific internal energy of gs, J/K_g
I_s	Specific internal energy of solids, J/K_g
i	Computing mesh index - r-direction
j	Computing mesh index - z-direction
K_b	Thermal conductivity of "packet", W/m°K
K_g	Thermal conductivity of gas phase, W/m°K
K_s	Thermal conductivity of solid phase, W/m°K
K_p	Thermal conductivity of particles, W/m°K
K_{kl}	Generalized drag coefficient for momentum transfer between phase k and phase l (gm/cm^{3}s)
N	Total number of phases

Symbol	Term
Nu	Nusselt number
n	time step index
P	Pressure (dynes/cm^2)
R	Gas constant ($cm^2/sec^2 {}^\circ K$)
R_{ek}	Reynolds number for particulate phase k, defined by equation 2.9
γ	Spacial coordinate (cm)
T	Temperature (°K)
t	Time(s)
U_{mf}	Mnimum Fluidization Velocity (cm/s)
u_k	Velocity of phase k in r direction (cm/s)
v_k	Velocity of phase k in z direction (cm/s)
V_k	Velocity vector of phase k (cm/s)
X_k	Fraction of solids phase k in a mixture of k and l, defined by equation 2.33
y	Spacial coordinate (cm)
z	Spacial coordinates (cm)

Greek Symbols

α	Correction factor used in equation 2.33
β	$\left(\frac{\alpha P}{\alpha D}\right)$ (cm^2/s^2)
δg	Convergence criteria (g/cm^3)
δt	Time step (s)
δr, δz	Mesh sizes (cm)
ε_k	Volume fraction of phase k
ε_{jl}	Solids volume fraction in a random mixture of solids phase k and l
ε_1	Void fraction

Symbol	Term
ε^*	Threshold void fraction below which solids stress is significant
ϕ_k	Volume fraction of solids phase k at maximum packing
ϕ_k	Characteristic shape factor of solids phase k
μ_1	Gas viscosity (gm/cms)
ω	Relaxation parameter
ρk	Density of phase k (gm/cm^3)
ρ_1	Density of the gas phase (gm/cm^3)
ρ_{sk}	Constant density of the solids phase (gm/cm^3)

ABSTRACT

General flow equations for the flow of gas and solid particles of different sizes and densities were developed. A collection of particles of the same size and density are considered to form a continuum — a particulate phase. Thus the flow of a mxiture of gas and particles is described using multiphase flow equations.

The empirical information used in the model are as follows. The gas-particle momentum transfer term used is a generalization of the correlation used by Gidaspow and Ettehadieh (1983) in their single particle size fluidized bed model. A generalization of the solids stress term used by them is also used to prevent the solids volume fraction from becoming unrealistically large. To describe the momemtum transfer between the particulate phases an approximate expression was derived. A general purpose computer program to solve the multiphase flow equations based on the K-FIX program (Rivard and Torrey, 1977) was developed. The model was applied to a variety of gas-solids flow problems involving one or two particle types.

The model was applied to the study of mixing in a fluidized bed by a bubble. The initial condition was that of two segregated layers of particles: the upper layer consisting of 230 μm particles of density 2.6 g/c_m^3 and the bottom layer consisting of 800 μm particles of density 2.91 g/c_m^3. At time zero a central jet was turned on and the grid flow was increased to keep the large particles at minimum fluidization. The computations, carried out up to 0.56 s, showed a bubble in the bottom layer and the mixing induced by the drift behind that bubble. The upper

layer expanded considerably as seen in an experiment. However, the computations did not go far enough to predict bubbles in the upper layer.

Energy equations were added to a single particle version of the computer model and heat transfer from bed to wall was studied. The large heat transfer coefficients characteristic of fluidized beds were computed without an enhancement of heat transfer by turbulence. They agreed with measurements reported by Ozkaynak and Chen (1980) within the accuracy of estimated thermal conductivity of solids.

A viscosity term was added to model radial velocity profiles in a pneumatic conveyor. The velocity profiles were similar to one of two contradicting sets of published experimental data. The computations predicted nearly flat velocity profiles. One-dimensional simulation to study segregation in a pneumatic conveyor for a binary mixture was carried out. The results were compared to experimental data.

Three multiphase flow models were studied to determine critical flow conditions. It was found that the dense phase transport may be limited by a critical velocity equal to $\sqrt{G/\rho_s}$ where G is an "elastic" modulus of the powder and ρ_s is its density.

CHAPTER I

INTRODUCTION

The modeling of fluidized beds using hydrodynamic equations started with the Davidson model in 1961. Davidson's model accurately predicted many features of the bubble motion in a fluidized bed. During the later years several investigators attempted hydrodynamic modeling of the fluidized bed (Jackson, 1963; Murray, 1965; Pigford and Baron, 1965; Soo, 1967; Ruckenstein and Tzeculescu, 1967). But as the models became more detailed and hence more complex, they also became analytically intractable.

The energy crisis in the 70's made it necessary to look at alternative sources of energy. This spurred interest in the use of fluidized bed gasifiers and combustors to utilize coal more efficiently and cleanly. The scale-up of these reactors to large scale units is difficult. This lead to the development of detailed numerical models for fluidized bed reactors by Systems, Science, Software (Scheneyer, et al., 1981) and JAYCOR (Scharff, et al., 1982). A parallel development at IIT, Chicago, produced a single particle model for an isothermal fluidized bed (Ettehadieh, 1983; Gidaspow and Ettehadieh, 1983). The model was based on the K-FIX program (Rivard and Torrey, 1977) developed at the Los Alamos Scientific Laboratory.

The attraction for a hydrodynamic model is due to its wider applicability (ie. the ease with which any restrictive assumptions may be lifted) and its ability to predict transient behavior of the local variables. It is hoped that these predictions may be applied fruitfully to improve practical designs. For example:

1. An understanding of the mixing of coal particles in a bed of limestone particles could lead to a better design for a coal injection system in a fluidized bed combustor.
2) It is necessary to know the motion of solid particles near a heat exchanger tube to study the corrosion/erosion rates.
3) The mechanism of bubble eruption at the bed surface could lead to a better understanding of the entrainment process.
4) It may be important to know the local heat transfer characteristics in some reactor applications.
5) Prediction of the relative velocities between particles may be useful in agglomeration/attrition studies.

It should be pointed out that hydrodynamic modeling is not without its inherent pitfalls and is not an alternative for careful experimentation. In fact due to the too few physical inputs a badly formulated problem could easily stray away from the reality and hence some experimental comparison is a must in the initial developmental stages of the model. The experimental efforts at IIT is a reflection of this necessity (Gidaspow, Lin and Seo, 1983).

The IIT model can predict the evolution, propagation and eruption of a bubble in a single particle fluidized bed as depicted in Figure 1.1 (Gidaspow, Syamlal, and Seo, 1985). The solids velocity vector plots for the same simulation are shown in Figure 1.2. Note that the model correctly predicts the solids circulation around the bubble and the powerful wake motion behind the bubble. The mechanism of solids transport due to the wake can be easily visualized.

In this thesis we intend to explore futher applications of the hydrodynamic model. In many practical applications of interest,the use

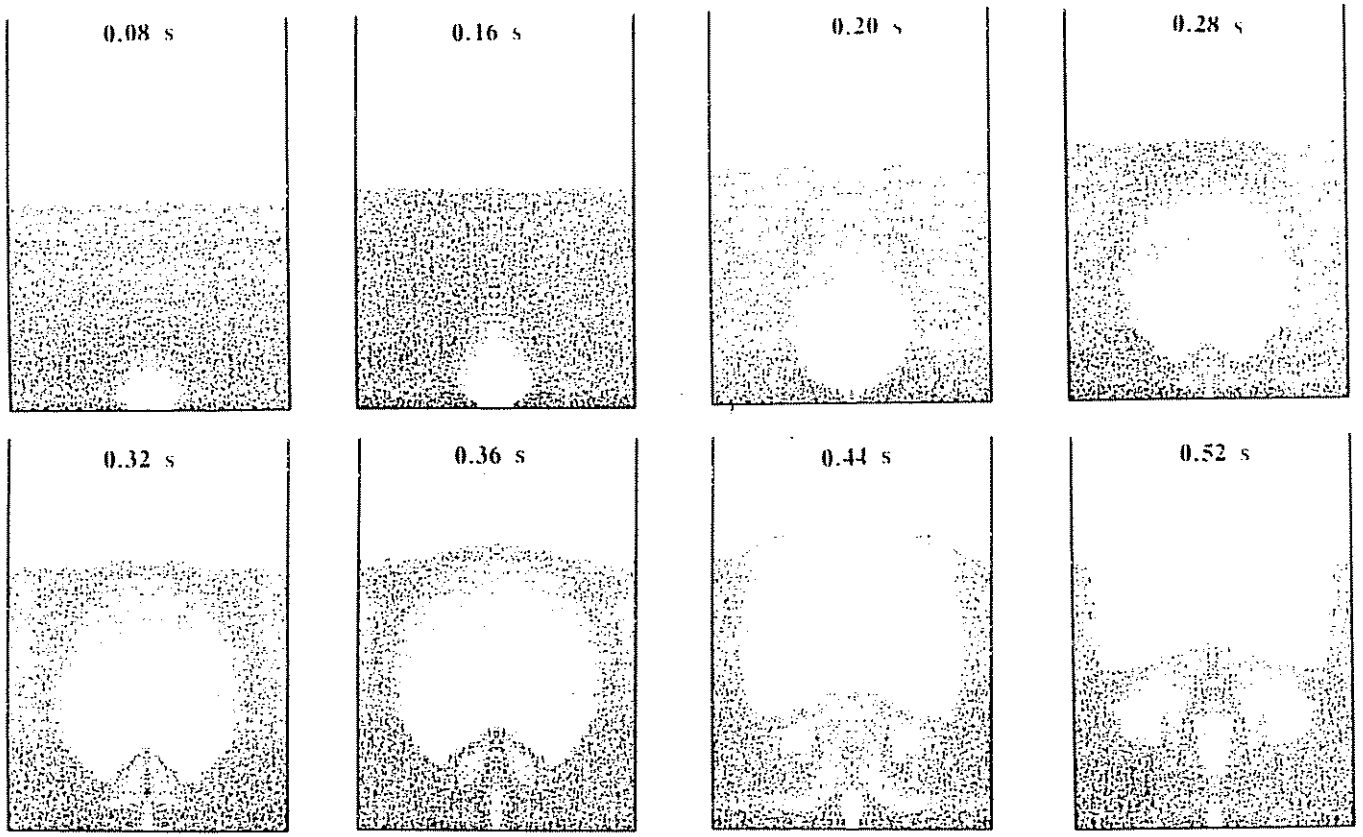


Figure 1.1. The Evolution and Propagation of a Bubble in a Bed of 800 μ_m Particles (jet velocity = 5 m/s)

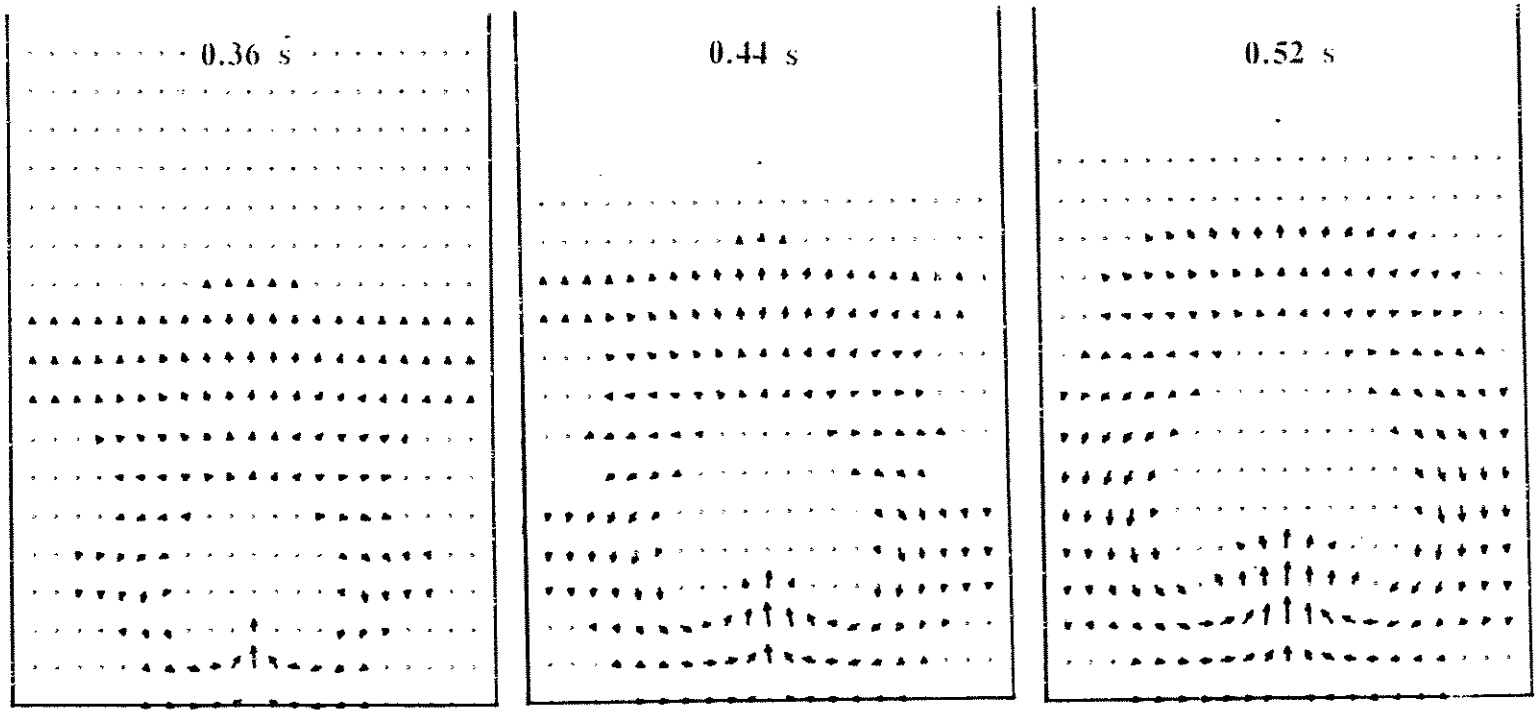


Figure 1.2. Solids Velocity Vector Plots for the Same Simulation as Shown in Figure 1.1.

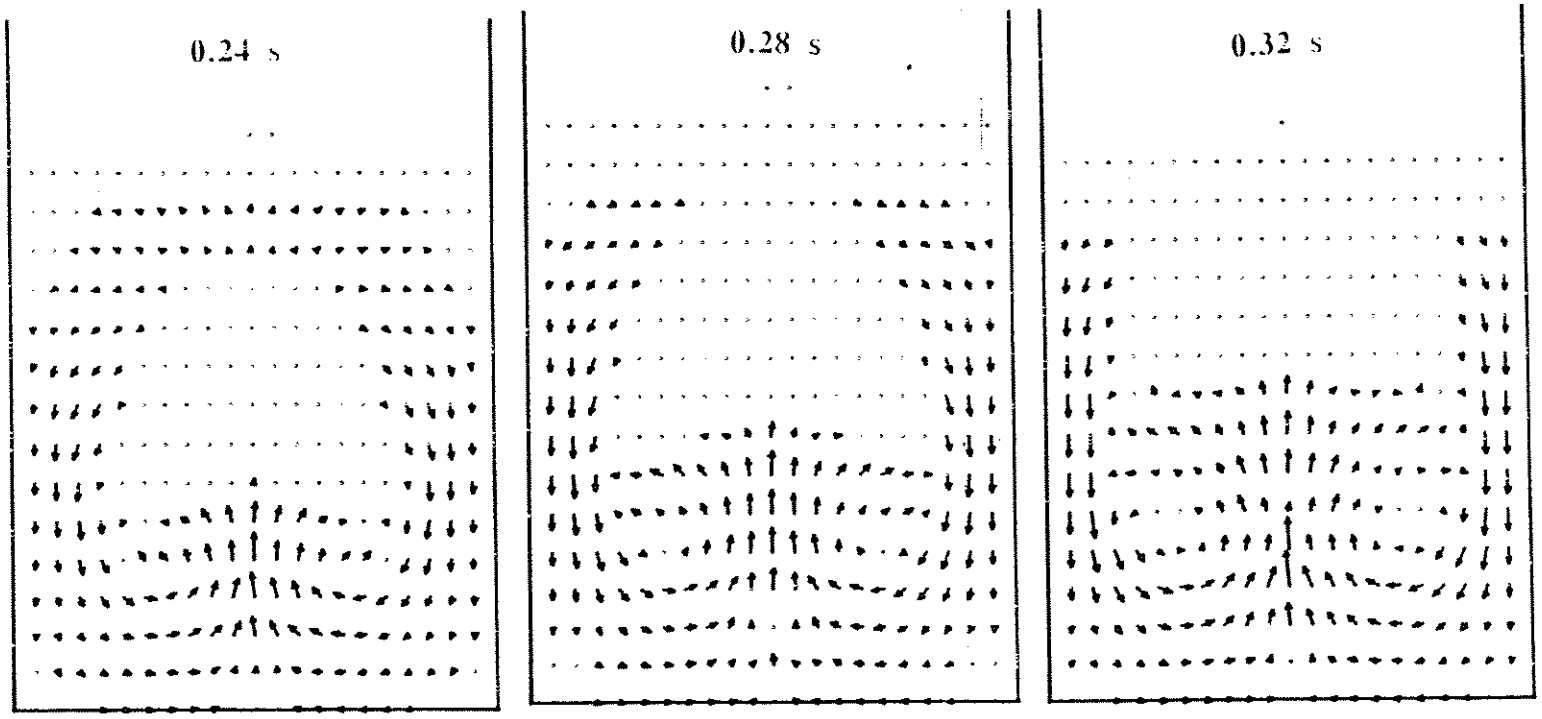


Figure 1.2. Solids Velocity Vector Plots for the Same Simulation as Shown in Figure 1.1.

of a single prarticle type may be too restrictive. Hence we extend the basic model using multiphase flow equations. Later on, the addition of energy equations and viscosity terms are considered.

In Chapter II the multiphase flow equations and the necessary constitutive relations are developed. Also included is a discussion of the initial and boundary conditions to be used. In Chapter III the model is used to study the mixing induced by a bubble in a segregated bed of two particulate phases. In Chapter IV energy equations are added to the basic hydrodynamic equations in an effort to check whether the model can predict the enhanced heat transfer rate in a fluidized bed. In Chpater V a viscosity term is added and the model is applied to study pneumatic conveying. In Chapter VI one of the interesting features of the equations, the prediction of a 'critical flow', is explored.

CHAPTER II

MULTIPHASE HYDRODYNAMICS

The Governing Equations

The central theme of this thesis is the application of multiphase flow equations to fluidization and pneumatic conveying. The equations and the necessary constitutive relations are developed in this chapter. These equations have been discussed and debated upon in several publications such as Soo (1967), Ishii (1975), Pai (1977), Lyczkowski, Gidaspow and Solbrig (1983).

The derivation of multiphase flow equations may be done using volume averaging of the single phase flow equations. It is necessary to make several drastic assumptions, eventually, to get the equations in a solvable form. The details of such a procedure are easily found in literature and are not presented here.

Here we will consider a multiphase system consisting of particles and a gas. The solid particles could be of different densities and diameters. Following a suggestion of Soo (1967) we consider that particles of identical densities and diameters form a continuum - a particluate phase. For example if we have a mixture of two types of particles, which differ in diameter, density or both diameter and density, we treat them as two distinct particlate phases.

In a practical situation, however, there is a distribution in the particle diameters and densities. Then it would be necessary to discretize such a distribution and consider it as several particulate

phases. Each of these particulate phases will be identified by some characteristic diameter and density, which are some average values of the discretized distribution.

The major weakness of the model lies in the necessity for discretization. When there is a very broad distribution in the particle properties, the number of particluate phases to be defined become very large for getting reasonable accuracy. Then the number of equations to be solved become intractably large (Pai, 1977). There are situations, however, such as segregation or entrainment in a fluidized bed where it is sufficient to study the dynamics of a binary or a tertiary mixture for a better physical understanding (Valenzuela, 1983). Thus, there is sufficient motivation for developing and utilizing a multiparticle model.

Actually the phases cannot coexist at the same spatial location. Hence we define a phase volume fraction ε_k for the phase "k" over a volume much larger than max $|d_k^3|$ but smaller than the volume of the computational region. The microscopic density may now be modified as

$$\rho_k' = \varepsilon_k \rho_k \tag{2.1}$$

Now the pseudo-fluids with the macroscopic densities ρ_k' are considered to form interpenetrating continua - one gas phase and possibly several particluate phases. Due to the presence of interfaces it is necessary to consider interphase mass, momentum and energy transport. Here we do not consider any interphase mass transport. The drag terms in the momentum equations given later account for the interphase momentum transport. The equations of motion may be derived just as in the case

of a single phase with suitable source terms such as drag terms. The set of equations used here for an isothermal case is given below. Note that the subscript "1" is used to indicate the gas phase.

Continuity Equations

$$\frac{\partial}{\partial t}(\rho_k \varepsilon_k) + \nabla \cdot (\varepsilon_k \rho_k V_k) = 0 \tag{2.2}$$

Momentum Equations

$$\frac{\partial}{\partial t}(\rho_k \varepsilon_k V_k) + \nabla \cdot (\varepsilon_k \rho_k V_k V_k) = -\varepsilon_k \nabla P + \rho_k \varepsilon_k g + G_k \nabla \varepsilon_1 + \sum_{\ell=1}^{N} K_{k\ell}(V_\ell - V_k) \tag{2.3}$$

where,

$$\sum_{k=1}^{N} \varepsilon_k = 1 \tag{2.4}$$

The gas phase is considerd to be ideal and the particulate phases are considered to be of constant densities.

$$\rho_1 = \frac{P}{RT} \qquad \rho_k = \rho_{sk} \quad \text{for } k \geq 2 \tag{2.5}$$

Gas-Particle Interaction

Gidaspow and Ettehadieh (1983) for their single particle model derived the gas-particle friction term from the Ergun equation and from Wen and Yu's (1966) correlation. These constitutive relations have been generalized as follows.

$$K_{1k} = K_{k1} = \frac{150\ (1 - \varepsilon_1)\ \varepsilon_k \mu_1}{\varepsilon_1 (d_k \phi_k)^2} + \frac{1.75\ \rho_1 |\ V_1 - V_k |\ \varepsilon_k}{(d_k \phi_k)} \quad 0.2 \leqq \varepsilon_1 < 0.8$$

$$= \frac{3}{4} C_{Dk} \frac{\varepsilon_k |\ V_1 - V_k |\ \rho_1 \varepsilon_k}{(d_k \phi_k)} - f\ (\varepsilon_1) \quad 0.8 \leqq \varepsilon_1 \leqq 1.0 \tag{2.6}$$

where,

$$f\ (\varepsilon_1) = \varepsilon_1^{-2.65}, \tag{2.7}$$

$$C_{Dk} = \frac{24}{Re_k} (1 + 0.15\ Re_k^{0.687}) \qquad Re_k < 1000$$

$$= 0.44, \qquad Re_k \geqq 1000 \tag{2.8}$$

and

$$Re_k = \frac{d_k\ |\ V_1 - V_k\ |\ \rho_1 \varepsilon_1}{\mu_1} \tag{2.9}$$

Suppose that we have a bed of identical particles. While using a multiparticle model we have a choice of treating this as a single particlulate phase of volume fraction ε_2 or as several particluate phases such that their volume fractions add up to ε_2. In the first case

we get a single momentum equation and in the latter case several momentum equations. The drag relations have been generalized in such a way that these momemtum equations correctly add up to the single momentum equation in the former case.

A check on such a generalization would be to compute the minimum fluidization velocity as predicted by these equations. However, it should be noted that there is no well defined U_{mf} in a multiparticle case as there is in the single particle case (Yang and Keairns, 1983). Our equations also indicate this. Consider equation 2.3 in one dimension at a "minimum fluidization" condition. We get

$$0 = - \varepsilon_1 \frac{dp}{dy} - \sum_{\ell = 2}^{N} K_{1\ell} V_1 - \rho_1 \varepsilon_1 g \qquad (2.10)$$

for the gas phase and

$$0 = - \varepsilon_k \frac{dp}{dy} + K_{1k} V_1 - \rho_k \varepsilon_k g \qquad (2.11)$$

for the particulate phases. Summing all these equations we get

$$- \frac{dp}{dy} = \sum_{k=1}^{N} \rho_k \varepsilon_k g \qquad (2.12)$$

which is the expected result that the weight of the bed is supported by the pressure drop. However, equations 2.10 and 2.11 form an over determined system for V_1. The reason for this is that there is an interaction between the particulate phases, which is missing in equation 2.11. These interaction terms are not easily defined and depend upon the flow history. For this reason, as shown in Figure 2.1, we observe a

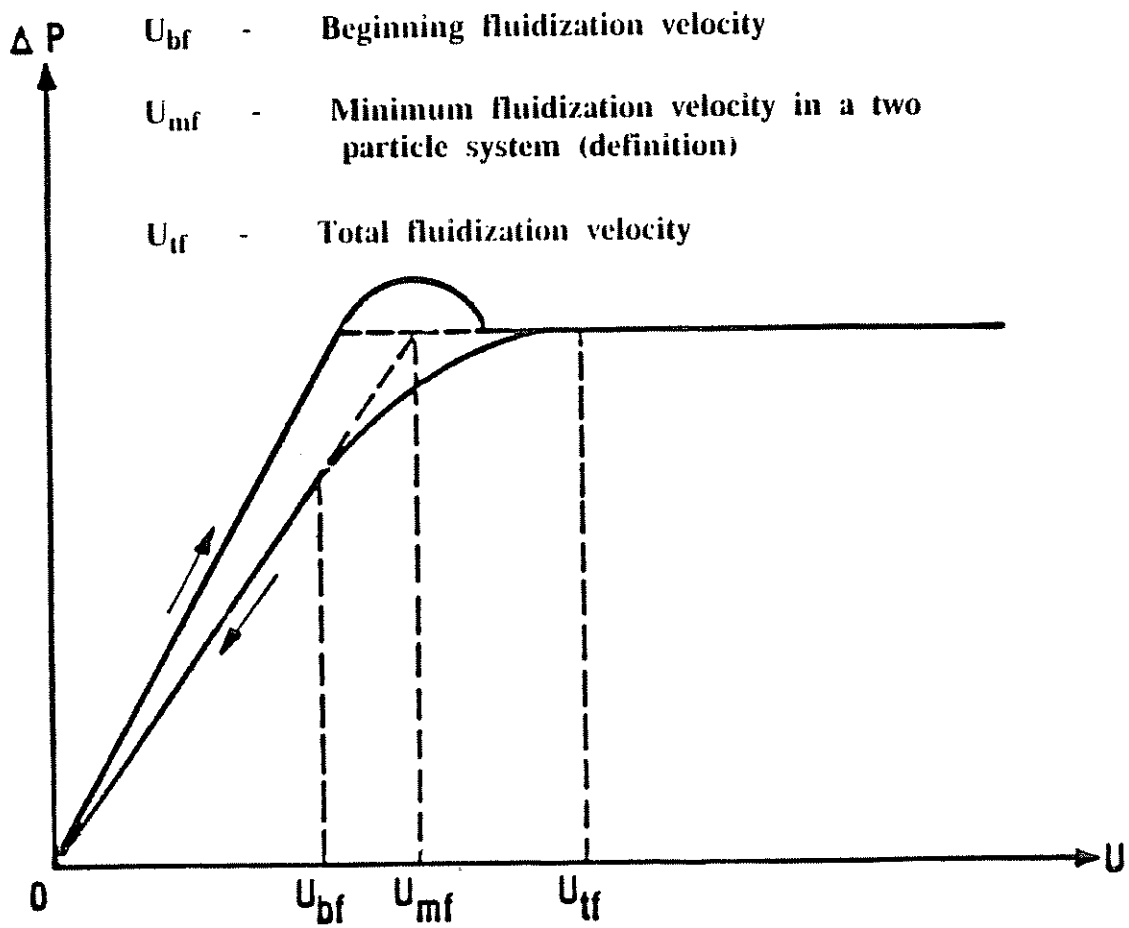


Figure 2.1. Pressure Drop Versus Gas Velocity for a Bed of Two Particle Size

hysteresis between the branch with increasing gas velocity and the branch with decreasing gas velocity (Yang and Keairns, 1983). Maintenance of the uniformity of the mixture and reproducibility of the results is poor (Rowe, and Nienow, 1975; Chiba etal., 1979). With this in mind we try some rough comparisons. We hope that 2.10 and 2.12 define the U_{mf} in some average sense as follows

$$\sum_{k=1}^{N} \rho_k \varepsilon_k g - \sum_{K=2}^{N} K_{1k} U_{mf} - \rho_1 g = 0 \qquad (2.13)$$

The U_{mf} calculated using equation (2.13) is compared to the experimental value and to the computed value using an equation due to Rowe and Nienow (1975) in Table 2.1.

Particle-Particle Interaction

A particle-particle friction term must be included in the equations to handle the momentum exchange between the particulate phases. Arastoopour, Lin and Gidaspow (1980) observed that such a term is necessary to give a correct segregation among particles of different sizes in a pneumatic conveyor. An equation for such an interaction in a dilute mixture has been suggested by Soo (1967). A similar expression was also used by Nakamura and Capes (1976). Greenspan and Ungarish (1982) did not use such an interaction term for the settling of a multiparticle mixture. Arastoopour, Wang and Weil (1982) experimentally studied such an interaction in a pneumatic conveyor.

Here we intend to derive an approximate expression to compute the momentum exhange between particulate phases in a dense bed. The task is to model the momentum exchange between phases "k" and "ℓ" given the local values of the field variables ε_k ε_ℓ, ρ_k,ρ_ℓ, d_k, d_ℓ and the

relative velocity $V_{k\ell}$ at an instant. We also assume that only binary interactions are important.

We use an approach similar to Bagnold (1954) to get the average separation between the particles of phase "k" and "ℓ" . Consider a collection of particles with modified diameters d_k^* and d_ℓ^*, such that they form a random close packed structure as shown in Figure 2.2.

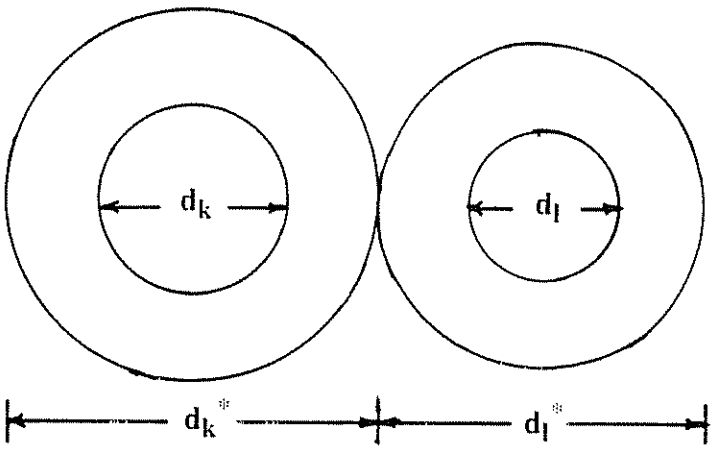


Figure 2.2. Definition Sketch for a Two Particle System

Table 2.1 Comparison of Minimum Fluidization Velocities of Multiparticle Systems

	Components		Experimental		Rowe and Nienow (1975)	Equation (2.13)
	d_k (μ_m)	Wt %	ε_{mf}	U_{mf} (cm/s)	U_{mf} (cm/s)	U_{mf} (cm/s)
	569	16.7				
Mixture	452	16.7				
A	284	25.0	0.355	2.59	2.33	1.85
			155	16.6		
			102	25.0		
Mixture	569	50.0				
B	284	50.0	0.394	9.14	12.08	17.59

The maximum solids volume fraction of a random close packed structure $\varepsilon_{k\ell}$ may be calculated as a function of d_k^*, d_ℓ^* and the composition using the following empirical formula due to Fedors and Landel (1979).

$$\varepsilon_{k\ell} = [(\Phi_k - \Phi_\ell) + (1 - a)(1 - \Phi_k)\Phi_\ell][\Phi_k + (1 - \Phi_\ell)\Phi_k]\frac{X_k}{\Phi_k} + \Phi_\ell \quad \text{for } X_k \leq \frac{\Phi_k}{\Phi_k + (1 - \Phi_k)\Phi_\ell}$$

$$= (1 - a)[\Phi_k + (1 - \Phi_k)\Phi_\ell](1 - X_k) + \Phi_k \quad \text{for } X_k \geq \frac{\Phi_k}{\Phi_k + (1 - \Phi_k)\Phi_\ell} \qquad (2.14)$$

where

$$a = \sqrt{\frac{d_\ell^*}{d_k^*}} \qquad (d_k^* \geq d_\ell^*) \qquad (2.15)$$

Φ_k, Φ_ℓ - Solids volume fraction at maximum packing in a single particle system for phases "k" and "ℓ" respectively.

$$X_k = \frac{\varepsilon_k^*}{\varepsilon_k^* + \varepsilon_\ell^*} \qquad (2.16)$$

Now

$$\varepsilon_k^* = \varepsilon_k \frac{(d_k + \lambda_k)^3}{d_k^3}$$

$$= \varepsilon_k \, (1 + s_k)^3 \tag{2.17}$$

where

$$s_k = \frac{\lambda_k}{d_k} \tag{2.18}$$

Similarly

$$\varepsilon_\ell^* = \varepsilon_\ell \, (1 + s_\ell)^3 \tag{2.19}$$

$$s_\ell = \frac{\lambda_\ell}{d_\ell} \tag{2.20}$$

To continue the derivation we have to make the assumption that

$$s_k = s_\ell = s \tag{2.21}$$

to get

$$a = \sqrt{\frac{d_\ell}{d_k}} \tag{2.22}$$

where we have assumed that $d_\ell \leq d_k$. Otherwise the indices "k" and "ℓ" should be interchanged in all the expressions. Also

$$X_k = \frac{\varepsilon_k}{\varepsilon_k + \varepsilon_\ell} \tag{2.23}$$

Now from equations 2.17, 2.19 and 2.21

$$\varepsilon_{k\ell} = \varepsilon_k^* + \varepsilon_\ell^*$$

$$= (\varepsilon_k + \varepsilon_\ell)\ (1 + s)^3 \qquad (2.24)$$

and

$$s = \left(\frac{\varepsilon_{k\ell}}{\varepsilon_k + \varepsilon_\ell}\right)^{1/3} - 1 \qquad (2.25)$$

giving the average distance between the particles as

$$\lambda_{k\ell} = \frac{\lambda_k + \lambda_\ell}{2} = \frac{(d_k + d_\ell)}{2}\ s$$

$$= \frac{(d_k + d_\ell)}{2}\ \left[\left(\frac{\varepsilon_{k\ell}}{\varepsilon_k + \varepsilon_\ell}\right)^{1/3} - 1\right] \qquad (2.26)$$

It should be pointed out that it is not certain whether the average collision distance as given by equation (2.26) is appropriate for all values of ε_k and ε_ℓ. In the limiting case of maximum packing, it does give the correct value of zero distance of separation. Later we will see that in the limit of ε_k, $\varepsilon_\ell \to 0$, also the expression for particle particle interaction reduces itself to Nakamura and Capes's (1976) equation for the dilute case.

Consider the collision of a pair of particles separated by a distance $\lambda_{k\ell}$ as shown in Figure 2.3.

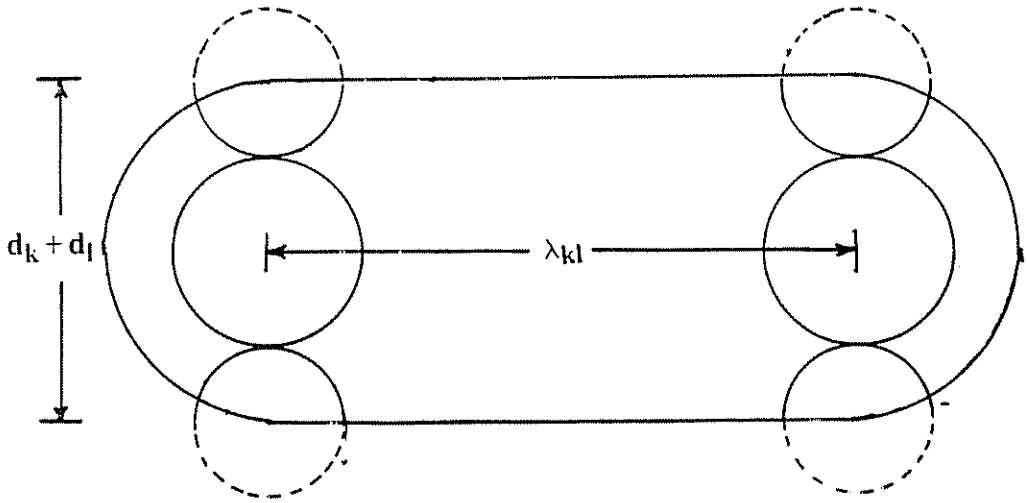


Figure 2.3 Definition Sketch for a Collision

In a volume ΔV the number density of "k" particles

$$= \frac{\Delta V \ \varepsilon_k}{\frac{\pi}{6} d_k^3} \tag{2.27}$$

Since the "k" particle has to move through an average distance of $\lambda_{k\ell}$ before encountering an "ℓ" particle, the average frequency of collision is given by

$$\frac{|V_{k\ell}|}{\lambda_{k\ell}} \tag{2.28}$$

The volume of collision per "k" particle is given by (Figure 2.3)

$$\frac{\pi}{6} (d_k + d_\ell)^3 + \frac{\pi}{4} (d_k + d_\ell)^2 \lambda_{k\ell} \tag{2.29}$$

From equations 2.27 and 2.29 the total collision volume

$$= \frac{\Delta V}{\frac{\pi}{6} d_k^3} \varepsilon_k \left[\frac{\pi}{6} (d_k + d_\ell)^3 + \frac{\pi}{4} (d_k + d_\ell)^2 \lambda_{k\ell} \right] \qquad (2.30)$$

Total number of collisions = Number of "ℓ" particles in the collision volume of '"k" particles (Equation 2.30)

$$= \frac{\Delta V}{\frac{\pi}{6} d_k^3} \varepsilon_k \left[\frac{\pi}{6} (d_k + d_\ell)^3 + \frac{\pi}{4} (d_k + d_\ell)^2 \lambda_{k\ell} \right] \frac{\varepsilon_\ell}{\frac{\pi}{6} d_\ell^3} \qquad (2.31)$$

Numbers of collision/units volume/unit time

$$= \frac{\text{total number of collisions}}{\Delta V} \frac{(V_{k\ell})}{\lambda_{k\ell}}$$

$$= \frac{\varepsilon_k \varepsilon_\ell}{d_k^3 \; d_\ell^3 \; \frac{\pi}{6}} (d_k + d_\ell)^2 \left[d_k + d_\ell + \frac{3}{2} \lambda_{k\ell} \right] \frac{| V_{k\ell} |}{\lambda_{k\ell}} \qquad (2.32)$$

Momentum transferred/ collision averaged over all angles of approach

$$= \alpha \; (1 + e) \frac{m_\ell \; m_k}{m_\ell + m_k} \; V_{k\ell}$$

$$= \frac{\alpha \; (1 + e) \; \frac{\pi}{6} d_k^3 \rho_k d_\ell^3 \rho_\ell}{(d_k^3 \; e_k + d_\ell^3 \rho_\ell)} \; V_{k\ell} \qquad (2.33)$$

where e is the coefficient of restitution and α is a correction factor to give the average value.

The total rate of momentum transfer between phase"k" and "ℓ" from equations 2.32 and 2.33

$$= \frac{\alpha\ (1 + e)\ \varepsilon_k \rho_k \varepsilon_\ell \rho_\ell\ (d_k + d_\ell)^2\ (d_k + d_\ell + \frac{3}{2}\lambda_{k\ell})\ |\ V_{k\ell}\ |\ V_{k\ell}}{\lambda_{k\ell}\ (\ d_k^3 \rho_k + d_\ell^3 \rho_\ell)} \qquad (2.34)$$

Using the value of $\lambda_{k\ell}$ we can deduce that the particle- particle friction coefficient is given by

$$K_{k\ell} = \frac{\alpha\ (1 + e)\ \varepsilon_k \rho_k \varepsilon_\ell \rho_\ell\ (d_k + d_\ell)^2 [1 + 3\ (\frac{\varepsilon_{k\ell}}{\varepsilon_k + \varepsilon_\ell})^{1/3}]\ |\ V_{k\ell}\ |}{2\ (\ d_k^3\ \rho_k + d_\ell^3\ \rho_\ell)\ [(\frac{\varepsilon_{k\ell}}{\varepsilon_k + \varepsilon_\ell})^{1/3} - 1]} \qquad (2.35)$$

Though the above expression was derived for the momentum exchange in a dense bed it can be easily verified that it tends to the expression given by Nakamura and Capes (1976) as $(\varepsilon_k + \varepsilon_\ell)$ tends to zero.

<u>The Solids Stress Term</u>

Giadaspow and Ettehadieh (1983) used a solids stress term in their equations to hope to make the characteristics real and thereby to make the problem well posed as an initial value problem. Physically such a term prevents the solids volume fraction from becoming unrealistically large.

To ensure that the momentum equations correctly add up, as in the case of gas-particle drag we have the following simple choices for generalizing the solids stress term for the case of more than one particulate phase:

1. $\varepsilon_k\ G_1\ (\varepsilon_1)\ \nabla\ \varepsilon_1$ (2.36)

2. $-\ G_2\ (\varepsilon_1)\ \nabla\ \varepsilon_k$ (2.37)

or a linear combination of the above two cases. Consider a region of large total solids volume fraction into or out of which a number of particulate phases are flowing. Both the terms given above will compute a force of large magnitude. However, only equation (2.36) guarantees the correct direction of the force ie. the direction of decreasing total solids volume fraction. Equation (2.37) may or may not give the correct direction depending upon the direction of the gradient of ε_k. Hence here we use equation (2.36) as the solids stress term. Later on we will see that even this choice of the solids stress term leads to certain difficulties in the treatment of boundary conditions. It seems that the correct direction should be given by the gradient of deviation from the maximum packed state

ie. $- \varepsilon_k G(\varepsilon_1) \nabla (\varepsilon_1 - \varepsilon^*)$ (2.38)

where ε^* is the void fraction at a maximum packing for the given composition of the particulate phases. The precise nature of ε^* being unknown, such a formulation is not attempted here.

We will use the following expression for $G(\varepsilon_1)$ which in the absence of any flow can predict a packed state of the particles with a very small void fraction gradient.

$$G = g_1 \exp [g_2 (\varepsilon^* - \varepsilon_1)] \qquad (2.39)$$

where

$g_1 = 1.5 \times 10^{-3}$ and $g_2 = 500$

and ε^* may be varied according to the void fraction at the maximum packing expected. A typical value would be 0.4.

The Initial and Boundary Conditions

The problem specification is complete with the prescription of appropriate initial and boundary conditions. Here we will discuss the general approach with reference to fluidized beds and pneumatic conveyors.

A natural initial condition for the fluidized bed is that of minimum fluidization state, where solids are not mixing and the gas is percolating at the U_{mf} . When there are more than one particulate phase and if all the particulate phases are well mixed a minimum fluidization condition at the mixture U_{mf} may be prescribed. If the particulate phases are segregated, U_{mf} for the smallest sized particles may be specified. In a pneumatic conveyor uniform gas flow with or without solids may be prescribed as an initial condition. In all the above cases the pressure distribution in the computational region is linear and should be prescribed accurately.

Correct specification of the boundary condition is very important. Improper handling can make the numerical solution unstable (Moretti, 1969). The theory of characteristics may be used to handle the boundary conditions correctly (Koo and Kuo, 1977; Gough, 1979; Chakravarthy, 1983). The boundaries we consider permit only one dimensional flow and hence for simplicity we look at the equations in one space dimension. For the case of a single particulate phase the characteristic equation is

$$\frac{\varepsilon_1 \rho_2}{C_g^2} (\lambda - V_1)^2 (\lambda - V_2)^2 - \varepsilon_1 \rho_2 (\lambda - V_2)^2$$

$$- \varepsilon_2\rho_1 (\lambda - V_1)^2 - G\, \varepsilon_2\varepsilon_1\left(\frac{(\lambda - V_1)^2}{\varepsilon_g^2} - 1\right) = 0 \qquad (2.40)$$

and for two particulate phases the characteristic equation is

$$\frac{\varepsilon_1\rho_2\rho_3}{C_g^2} (\lambda - V_1)^2 (\lambda - V_2)^2 (\lambda - V_3)^3 - \varepsilon_1\rho_2\rho_3 (\lambda - V_2)^2 (\lambda - V_3)^2$$

$$- \varepsilon_2\rho_1\rho_3 (\lambda - V_1)^2 (\lambda - V_3)^2 - \varepsilon_3\rho_1\rho_2 (\lambda - V_2)^2 (\lambda - V_1)^2$$

$$- \frac{\varepsilon_1\varepsilon_3\rho_2 G}{C_g^2} (\lambda - V_1)^2 (\lambda - V_2)^2$$

$$- \frac{\varepsilon_1\varepsilon_2\rho_3 G}{C_g^2} (\lambda - V_1)^2 (\lambda - V_3)^2 + \varepsilon_1\varepsilon_3\rho_2 G (\lambda - V_2)^2$$

$$+ \varepsilon_1\varepsilon_2\rho_3 G (\lambda - V_3)^2 = 0 \qquad (2.41)$$

For the fluidized bed as well as the pneumatic conveyor at the inflow and outflow boundaries

$$\varepsilon_1 \to 1.0 \; ; \; \varepsilon_2, \varepsilon_3 \to 0.0$$

and $G \to 0.0$

Then equations 2.40 and 2.41 give the characteristic directions as

$$\lambda_{1,2} = V_1 \pm C_g \qquad (2.42)$$

$$\lambda_{3,4} = V_2 \qquad (2.43)$$

$$\lambda_{5,6} = V_3 \tag{2.44}$$

The following intative interpretation seem to be in order. Equation 2.42 shows that the characteristic directions are similar to those of the gas dynamics equations. The disturbances propagate at nearly the sonic velocity. Equations 2.43 and 2.44 say that the solids volume fraction and the solids velocity propagate with the respective solids velocity in a dilute mixture. Physically this makes excellent sense.

It is known that the number of boundary conditions necessary at any boundary is equal to the number of characteristics issuing from that boundary and entering the domain of interest (Lax, 1954). Hence we may summarize the necessary boundary conditions as follows.

1. Inflow boundary

In a fluidized bed zero solids velocity at the inlet approximately describes the contact discontinuity above the distributor plate. In a pneumatic conveyor the solids velocity and volume fraction should be specified.

In both the fluidized bed and the pneumatic conveyor for gas flow only one condition may be specified, since only one of the characteristics point into the computational region. An appropriate specification is a constant mass flow rate such as

$$\rho_1 V_1 = \text{const.} \tag{2.45}$$

Moretti (1969) criticizes the use of such a constant value at subsonic boundaries. However, here this is the only reasonable way of handling the inflow boundary, since the average flow rate may be experimentally measured.

2. Outflow boundary

For the solids it is not necessary to specify any condition unless the solids velocity become negative. For the gas flow, again there is only one characteristic pointing inwards. Prescription of constant pressure is an appropriate choice since we consider only subsonic flows.

3. Solid wall

At a solid wall we know for certain that the normal velocities are zero. Thus the task here is to ascertain whether it is consistent with the mathematical equations. For simplicity consider only one particulate phase. Now setting $V_1 = V_2 = 0$ in equation 2.40 we get

$$\frac{\varepsilon_1 \rho_2}{C_g^2} \lambda^4 - \left(\varepsilon_1 \rho_2 + \varepsilon_2 \rho_1 + \frac{G\ \varepsilon_2 \varepsilon_1}{C_g^2}\right) \lambda^2 + G\ \varepsilon_2 \varepsilon_1 = 0 \qquad (2.46)$$

Since we have to prescribe two conditions at the boundary we should have two +ve roots and two -ve roots for this fourth order equation. Thus the necessity for a non-zero solids stress term, G, is evident from equation 2.46. If G was zero two of the roots will vanish and we get only one +ve root which is incompatible with our initial assumption. Further to get the required roots, then we should have, two +ve roots for $s = \lambda^2$ which implies that

$$\varepsilon_1 \rho_2 + \varepsilon_2 \rho_1 + \frac{G\ \varepsilon_2 \varepsilon_1}{C_g^2} > \frac{2\varepsilon_1}{C_g} \sqrt{\rho_2 \varepsilon_2 G} \qquad (2.47)$$

It can be easily verified that equation (2.47) is always true.

Letting $V_1 = V_2 = V_3 = 0$ in equation 2.41 immediately shows that two of the λ's are zero which imply that we can get only two +ve roots at most. This is clearly an incompatibility due to the inaccurate formulation of the solids stress term. We can live with the existing formulation provided that in the wall region the solids are not well mixed.

CHAPTER III

MIXING OF PARTICLES IN A FLUIDIZED BED

An understanding of the mixing in a fluidized bed is of great importance in many applications of the fluidized bed. For example, in a fluidized bed combustor, it is important to know how coal particles mix in a bed of limestone particles. This is necessary for a proper design of the coal injection system. For this reason numerous studies have been reported in the literature on the mixing in a fluidized bed. A review of this literature has been given by Rowe and Nienow (1976) and Valenzuela (1982).

The models for mixing and segregation are mainly diffusion models (Kunii and Levenspiel, 1969; Shi and Fan, 1984) or models which consider the convective contribution (Rowe, Nienow and Agbim, 1972; Gibillaro and Rowe, 1974). It is evident that the single most important controlling factor for mixing is the bubble motion (Rowe and Nienow, 1976). Here we intend to study the mixing caused by a bubble using the hydrodynamic model developed in the previous chapter. The motivation for this being the wider applicability of hydrodynamic models and the advantage that these models may be extended to practical situations involving chemical reactions.

A computer code was developed based on the equations presented in the previous chapter. The method is essentially an extension of the K-FIX algorithm (Rivard and Torrey, 1977). The details of the numerical method and a user's manual are given in Appendix A. The model was applied to several hypothetical problems. Some of the results obtained are presented in Appendix B. In this chapter we present the results for

the simulation of mixing of two types of particles in a fluidized bed. These results have been compared to some limited experimental data. The properties of the two particle types used are presented in Table 3.1.

Table 3.1. Simulation Conditions for a Two Particle System

	d_p (μm)	ρ_p (g/cm^3)	U_{mf} (cm/s) Experiment	U_{mf} (cm/s) Using Equation 2.13
Ballotini	820	2.94	50.6	63.4
Glass beads	241	2.42	6.1	7.6

The geometry of the fluidized bed system is shown in Figure 3.1. The initial condition used was that of two separated layers of particles with the denser (Ballotini) particles at the bottom. Air flowing at a velocity of 12.2 cm/s kept the top layer of finer particles (glass beads) in a fluidized state. At time zero a central jet was turned on at a velocity of 5 m/s and the grid flow was increased to 50.6 cm/s. The boundary conditions are as given below.

<u>B.C.: 1</u> at $y = 0$

$\varepsilon = 1.0$

$V_1 \rho_1$ = Constant; i.e. a specified constant mass flux of gas through the jet and the grid region.

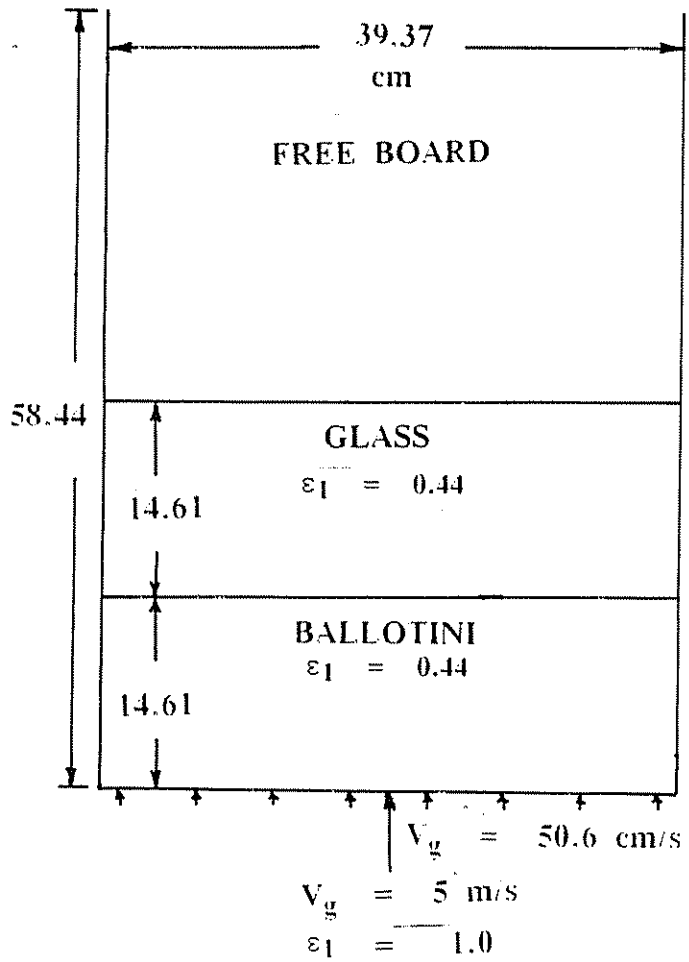


Figure 3.1. Geometry of the System for the Mixing Experiment

<u>B.C.: 2</u> at $y = L$

P = atmospheric; i.e. the bed is open to atmosphere.

<u>B.C.: 3</u> at $x = 0$

$U_1 = U_2 = U_3 = 0$; i.e. symmetry was assumed at the the center.

<u>B.C.: 4</u> at $x = W$

$U_1 = U_2 = U_3 = 0$; i.e. zero normal velocity due to the the presence of the wall.

The simulation was carried out for the passage of one bubble through the bed. The intention was to study the behavior of the bubble as it crosses the boundary between the denser and lighter particles. Some limited experimental data in terms of photographs of the bed in operation and in a collapsed state at the end of 0.5 s of operation are available. The denser particles were dyed blue to distinguish the different particulate phases. The volume fraction data of the two particulate phases at equal intervals of time were converted into a series of density plots and these plots were studied as a movie. Differently colored dots were used to distinguish between the two particulate phases. The bubble induced mixing can be clearly seen in the movie. A few frames from the movie are presented here. Here black dots are used in the density plots and two separate sets of plots are used for the two different particulate phases.

Figures 3.2a-f show the density plots for the volume fraction of the denser particles. The formation of a bubble in the bottom layer (of denser particles) and its propagation can be easily seen. The characteristic sharp boundary at the lower part of the bubble and the more diffuse boundary at the upper part can be easily seen. But as the

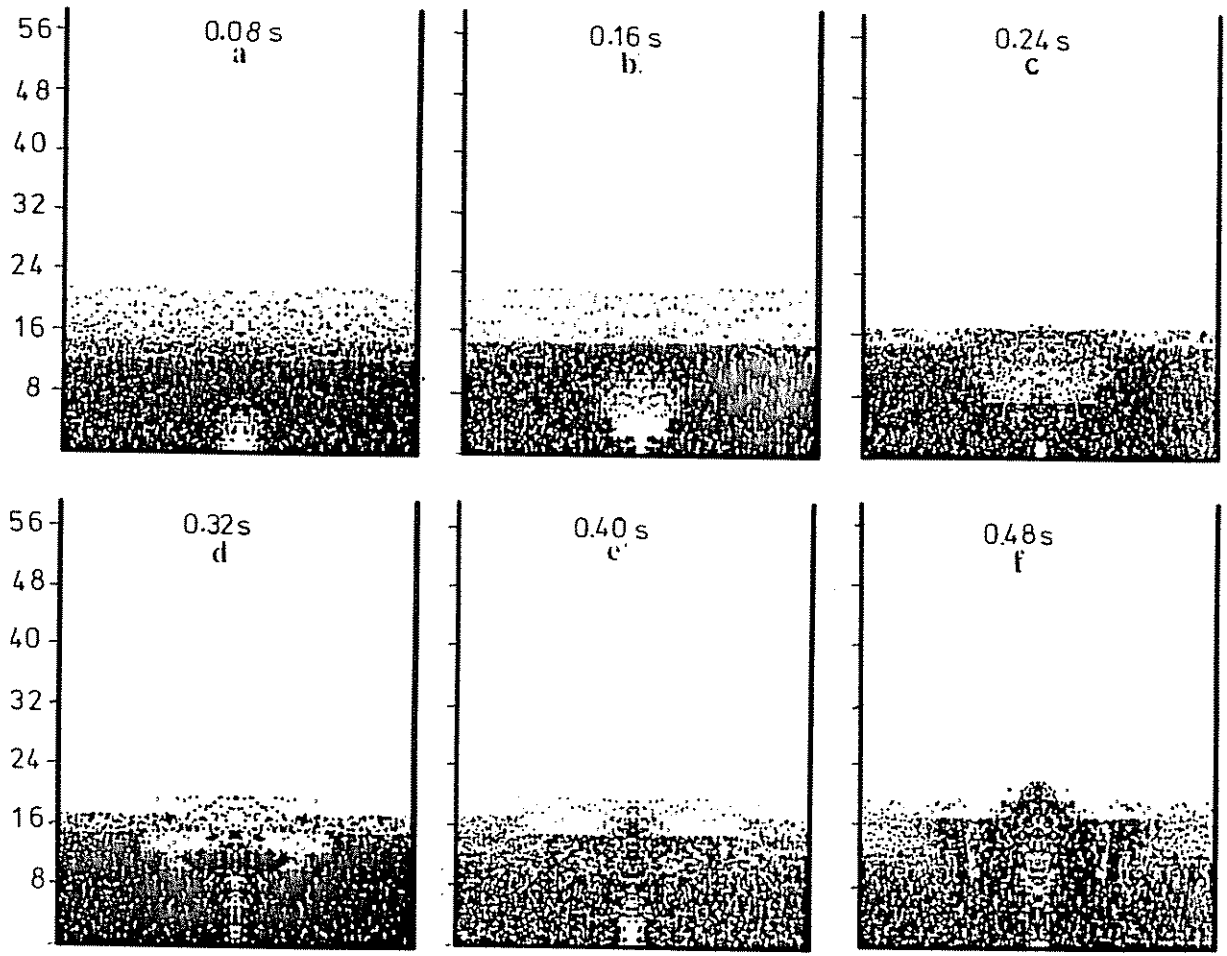


Figure 3.2. Volume Fraction Plots for the Ballotine Particles at Various Times

bubble reaches the interface between the particulate phases, the bubble loses its sharp boundaries and flattens out. The denser particles are not pushed into the lighter particle layer by the upper surface of the bubble. The denser particles drain back into the bottom layer. In a single particle case also this is true as it is evident from the pictures given by Rowe et al. (1965). Unlike the single particle case, the bubble is not well defined in the upper layer of lighter particles (See Figures 3.3 a-e). Also we do not see the transport of the wake phase of dense particles into the upper layer. In a photograph of the experiment, such a wake phase could not be detected. Beeckmans et al. (1985) report that when the density difference is sufficiently large almost complete shedding and replacement of the wake may occur just above the density discontinuity. After the bubble leaves the lower layer of the bed the formation of a drift of dense particles, behind the bubble can be seen. In Figure 3.4 the shape of the drift layer predicted by the simulation has been compared to experimental data. In the experimental case the bed was collapsed at the end of 0.5 s.

Figures 3.3 a-e show the behavior of the lighter particles (upper portion). The predominant phenomenon in this case is that of bed expansion. Since the average velocity in the bed is roughly eight times the U_{mf} of the lighter particles, there is considerable expansion in this layer. Figure 3.5 shows the experimental data of the interfaces after the first bubble leaves the bottom layer of dense particles. The average bed expansion data are compared in Table 3.2.

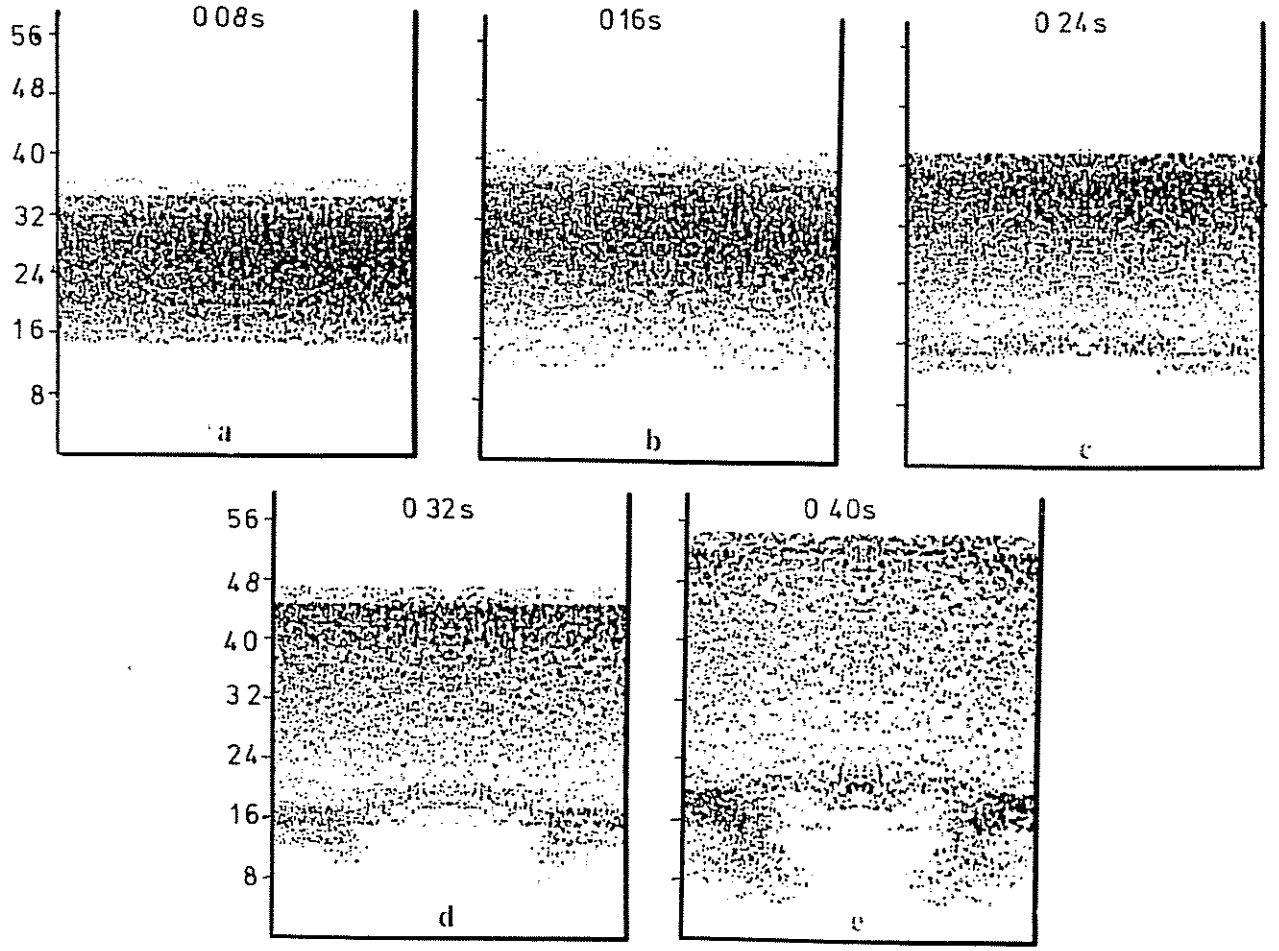


Figure 3.3. Volume Fraction Plots for the Lighten Particles at Various Times

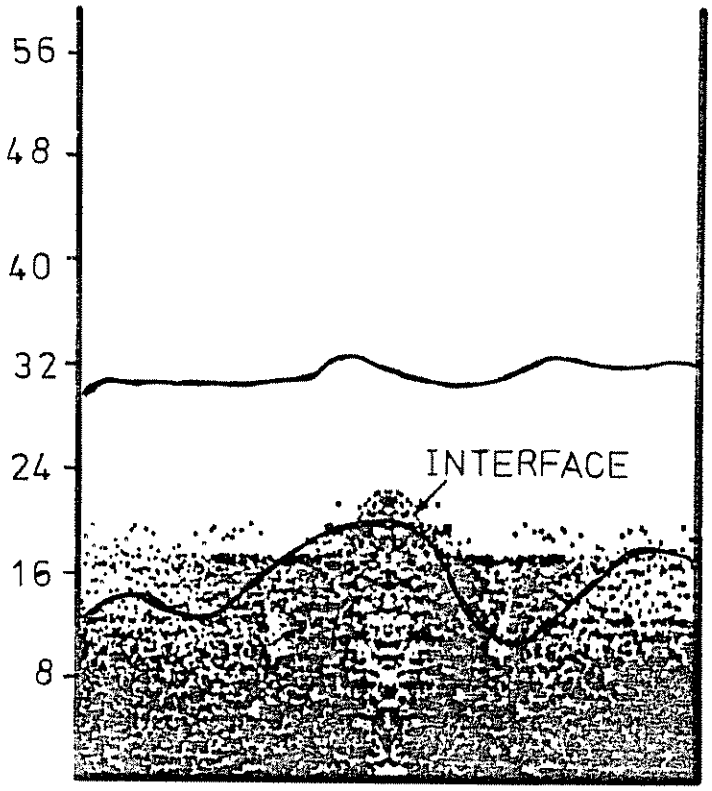


Figure 3.4. Comparison of the Drift Layer as Predicted by the Theory to Experimentally Observed Interface When the Bed is Collapsed at the End of 0.5s

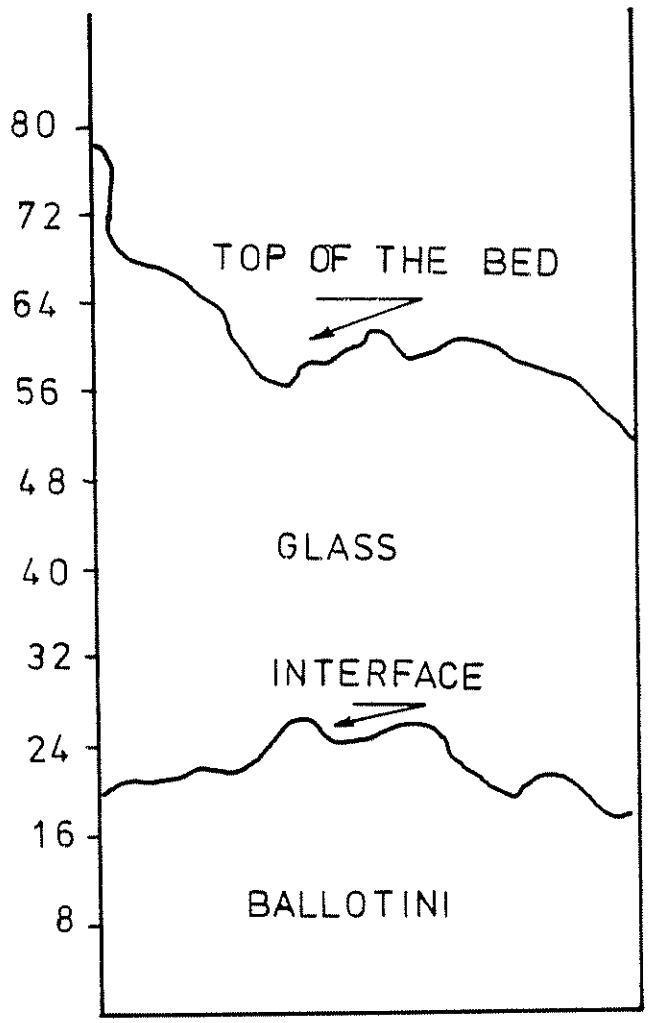


Figure 3.5. Experimentally Observed Interfaces When the Bed is in Operation

Table 3.2. Comparison of Fractional Bed Expansion Data

Particle Type	Bed Expansion Experiment	Bed Expansion Theory
Ballotini	1.48	1.18
Glass Beads	2.51	2.47

No well defined bubble can be detected in the upper layer. In the movie of the simulation a slug formation at the interface could be seen. The mixing mechanism as predicted by the theory is clear from Figures 3.2f and 3.3d-e. After the bubble leaves the dense particle layer and as the drift forms behind the bubble, the denser particles move towards the center into the drift layer. These particles are replaced by the descending lighter particles away from the center. This seems to be the mechanism of mixing in this case. The lighter particles can be seen to concentrate away from the central jet region, due to the interaction with the denser particles. Although this trend is apparent in the experimental data (See Figure 3.5), firm conclusions can be drawn only from precise measurements of the volume fractions in this mixing layer. This would be essential for a full verification of the constitutive relation used here for the particle-particle interactions. The simulation indicates a convenient point where the transient volume fractions could be measured for a more detailed comparison.

Figures 3.6a-e show the gas velocity vector plots at various times. The typical slow bubble profile in which the gas takes a short-cut through the bubble can be seen in the bottom layer. After the passage of the first bubble, a small second bubble can also be seen. Again, there is no evidence of a bubble in the upper layer.

Figures 3.7a-e show the dense solids velocity vector plots at various times. Initially a small amount of the dense particles get mixed with the lighter particles and hence are carried upwards. They eventually get well mixed with the light particles to form a very dilute mixture ($\varepsilon_1 \approx 0.002$). The bubble and the circulation of solids in its wake and the relatively stagnant regions near the side walls are also clear from these plots.

Figure 3.8a-e show the light solids velocity vector plots at various instants. As already seen in the density plots here also we can see the rapid expansion of the upper layer and can see that the solids slow down after the expansion is complete. In Figure 3.8e a layer of the particles can be seen being sucked down. Figures 3.8d-e show further mixing. The lighter particles can be seen to stay away from the central jet region and the stagnant region of dense particles near the walls.

Thus in this chapter we have demonstrated the possibility of studying the mixing process in a fluidized bed using a multiparticle model. Though not fully experimentally verified the model is capable of predicting the mechanism of the mixing process.

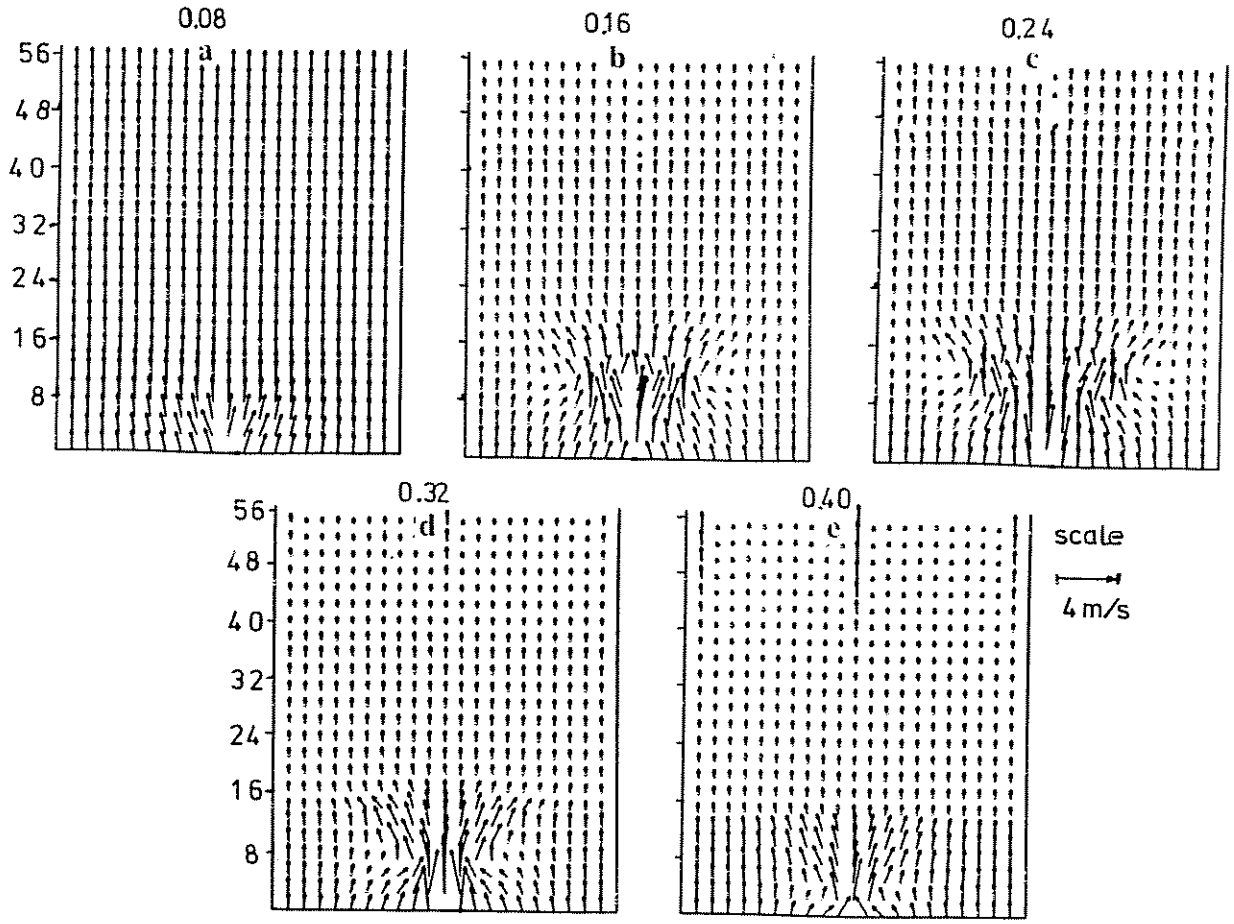


Figure 3.6. Gas Velocity Vector Plots at Various Times

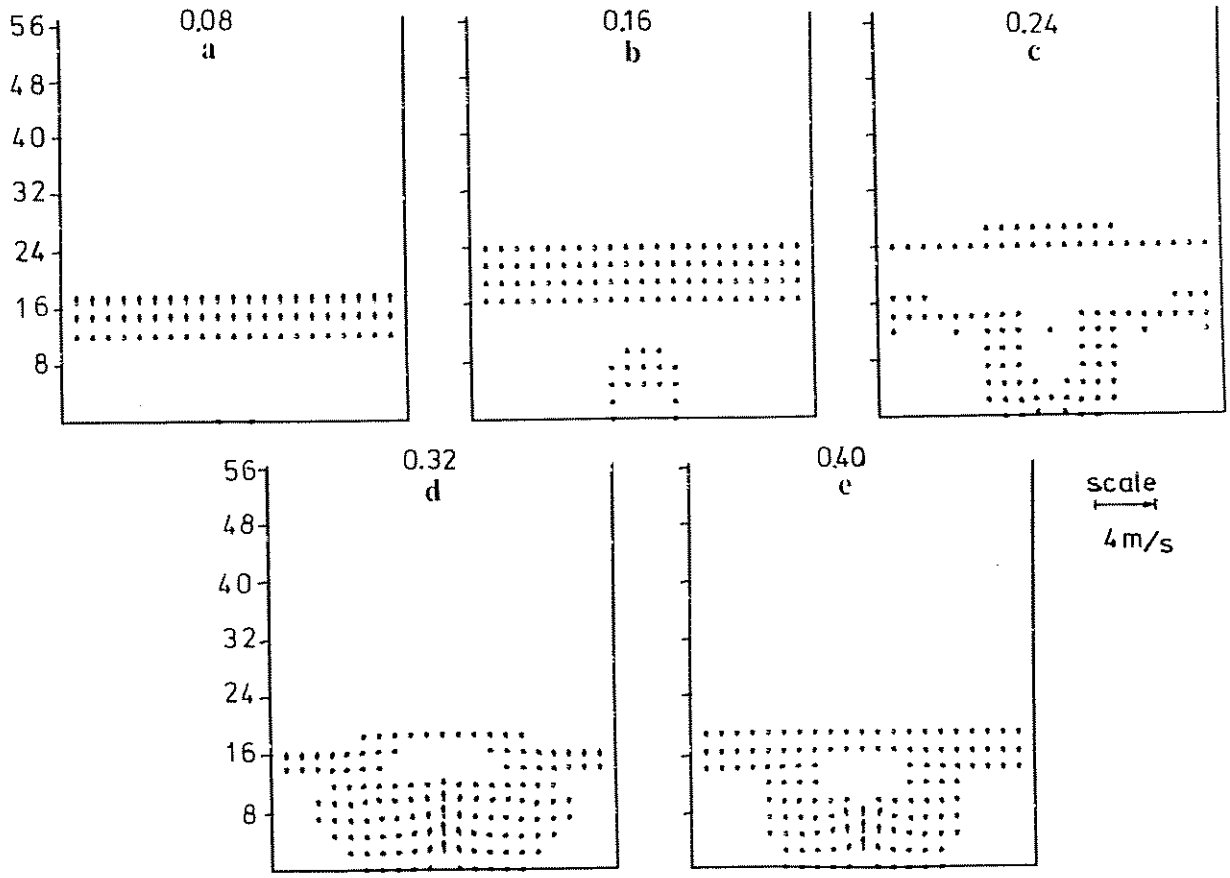


Figure 3.7. Solids Velocity Vector Plots of the Dense Bottom Layer

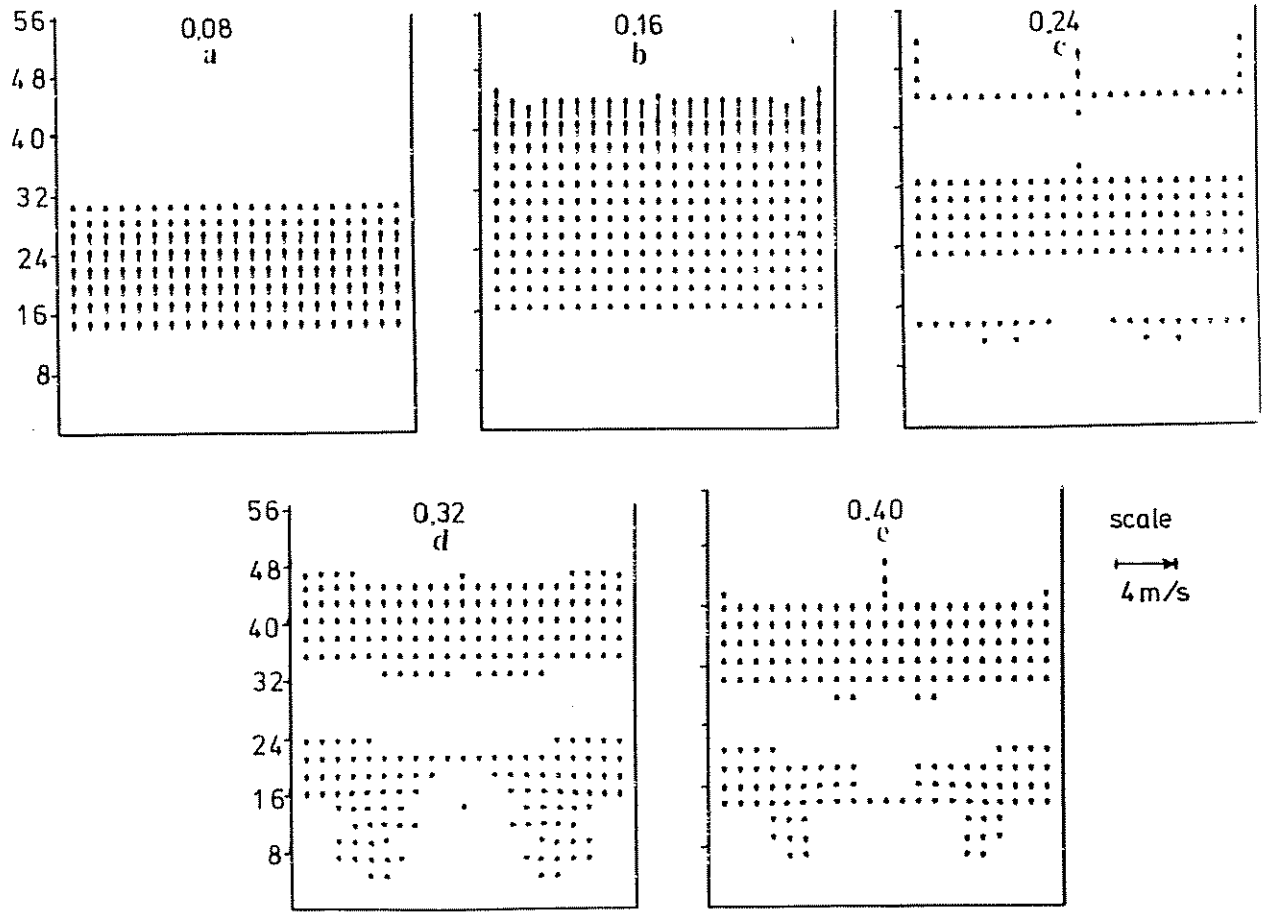


Figure 3.8. Solids Velocity Vector Plots for the Top Layer of Light Particles

CHAPTER IV

HEAT TRANSFER FROM A FLUIDIZED BED

Fluidized beds find application as excellent heat transfer media. The transition to fluidized state is accompanied by a remarkable increase in the bed to wall heat transfer coefficient. Such a drastic change is a result of the peculiar hydrodynamics in the fluidized state. Thus it has been a subject of intense investigation for the past several years. A review of these studies may be found in Gelperin and Einstein (1971), Botterill (1975) and Saxena et al. (1978).

The complexity of the heat transfer phenomena in the fluidized state is attested by the large number of correlations available for computing the heat transfer coefficient. But these formulae are valid only within the limits of the experimental conditions on which they are based, and may differ by almost two orders of magnitude from the actual coefficients in some cases (Gelperin and Einstein, 1971). Hence, the investigators are motivated to study the mechanism of heat transfer more closely.

From the early 50's investigators started proposing mechanistic models for predicting the heat transfer coefficient. As pointed out by Gelperin and Einstein (1971), these models may be broadly classified as follows, on the basis of the dominating mechanism:

1. Models based on conductive heat transfer through the fluid boundary layer near the heat transfer surface. These models have the major drawback of neglecting the thermal-physical properties of the solid.

2. Models using the assumption that the dominant part is played by the solids, in the transferring the heat. These models are unsuitable for gas fluidized beds.
3. Models based on the postulate that heat is transferred by "packets" of solids which are periodically replaced from the heat transfer surface by gas bubbles. This mechanism breaks down for ε greater than 0.7.

The last mechanism originally due to Mickley and Fairbanks has been extensively used in various modified forms. Later in this work we will find that the "packet" model is similar to the continuum approach developed here, in some respects.

The packet model gives the local instantaneous heat transfer coefficient in terms of a penetration mechanism as:

$$h_i = \left(\frac{k_a \rho_b C_s}{\pi \tau_p}\right)^{1/2} \qquad (4.1)$$

Here, "τ_p" is the time for which the packet was in contact with the heat transfer surface. Although, such detailed models describe the means by which surface to bed heat transfer takes place, they are of limited application because they require a knowledge of parameters which are not easily available (Botterill, 1975, Ozkaynak and Chen, 1980).

For large particle fluidized beds, a gas convection model of heat transfer was proposed by Adams and Welty (1979). In their work, the heat transfer from a horizontal tube is modeled using boundary layer equations, for the gas phase. The average interstitial gas velocity is calculated from an approximate analytical solution of the two-phase flow equations. The mechanism of heat conduction to the solid phase is

neglected. A transfer mechanism by means of turbulence is introduced. Such a mechanism was earlier used by Galloway and Sage (1970). The particle sizes used in both of the studies mentioned above are about an order of magnitude larger than those considered here.

A critical test for the hydrodynamic model developed here would be its ability to predict the enhanced heat transfer rate in a fluidized bed as discussed above. Here we use the multiphase flow equations discussed in Chapter II for a single particle size, and add energy equations as discussed below.

Energy Equations

The energy equations are written in a form found in the K-FIX computer code (Rivard and Torrey, 1977) as given below:

$$\frac{\partial}{\partial t}(\varepsilon\rho_g I_g) + \frac{\partial}{\partial x}(\varepsilon\rho_g I_g U_g) + \frac{\partial}{\partial y}(\varepsilon\rho_g I_g V_g)$$

$$= -p\left[\frac{\partial\varepsilon}{\partial t} + \frac{\partial}{\partial x}(\varepsilon U_g) + \frac{\partial}{\partial y}(\varepsilon V_g)\right] +$$

$$h_v(T_s - T_g) +$$

$$\frac{\partial}{\partial x}\left(K_g\varepsilon\frac{\partial T_g}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_g\varepsilon\frac{\partial T_g}{\partial y}\right) \qquad (4.2)$$

$$\frac{\partial}{\partial t}[(1-\varepsilon)\rho_s I_s] + \frac{\partial}{\partial x}[(1-\varepsilon)\rho_s I_s U_s] + \frac{\partial}{\partial y}[(1-\varepsilon)\rho_s I_s V_s]$$

$$= -P\left\{\frac{\partial}{\partial t}(1-\varepsilon) + \frac{\partial}{\partial x}[(1-\varepsilon)U_s] + \frac{\partial}{\partial y}[(1-\varepsilon)V_s]\right\}$$

$$+ h_v(T_g - T_s) \qquad (4.3)$$

$$+ \frac{\partial}{\partial x} [K_s(1 - \varepsilon) \frac{\partial T_s}{\partial x}] + [K_s(1 - \varepsilon) \frac{\partial T_s}{\partial y}]$$

These equations can be derived from the single phase equations using the volume averaging technique and approximating some of the resultant surface integrals. Note that the two phase constitutive equations are different from the single phase constitutive equations. The gas phase being continuous, its constitutive equations need not be modified. The gas phase is considered to be ideal and its specific heat and thermal conductivity are considered to be constants. For the particulate phase, the specific heat is the same as that for the material of the particles. However, particulate phase heat transfer, and interphase heat transfer need further considerations.

Boundary and Initial Conditions

Refer to Figure 4.1 for bed dimensions, the particle diameter, and other data used in the numerical calculations reported here.

The boundary conditions (B.C.) used are as follows:

B.C.1: At $y = 0$ we have

$\rho_g V_g = C_j$, that is we prescribe a constant gas mass flux through the distributor plate. $T_g = 298°K$.

B.C.2: At $t = L$, we have P = atmospheric and $V_s = 0$, that is, we have a wire mesh to prevent solid carry-over from bed.

B.C.3: At $x = 0$, $U_s = U_g = 0$, and $T_w = 298°k$.

B.C.4: At $x = W$, $U_s = U_g - 0$ and $T_w = 373°k$.

The initial condition is assumed to be that of minimum fluidization and the solids and gas at a uniform temperature of 298°k.

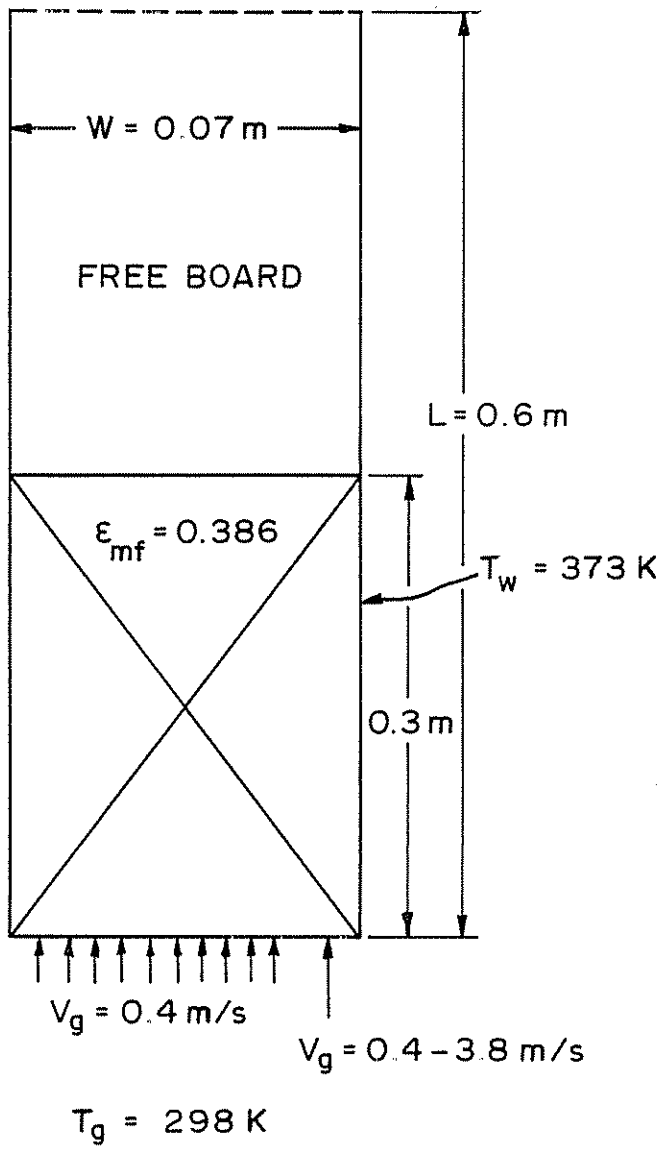


Figure 4.1. Typical Data For Numerical Computations

A free board of the same size as the initial bed height is provided to allow for bed expansion. A floating size free board has also been used. At time zero the gas flow is increased.

Interphase Heat Transfer

The interphase heat transfer term is written in a form h_v (T_g - T_s). The heat transfer occurs primarily through a gas film surrounding the individual particle. We have to use empirical correlations for predicting h_v. There is a large number of correlations available for packed bed gas-solid heat transfer. Some of them may be adapted for fluidized bed conditions. Three correlations from Zabrodsky (1966), Gelperin and Einstein (1971), and Wakao et al. (1979) are shown in Figure 4.2 for comparison. The former two correlations are suitable for fluidized bed interphase heat transfer. However, these correlations are known to differ from certain experimental data by 100 to 200%. In this study we use correlations taken from Zabrodsky as shown in Table 4.1. Also, the interfacial area needed from computing h_v is calculated based on the surface area of equally sized spherical particles. It is known that if a hot gas is injected into a cold bed, the gas is quenched by the time it flows past the first row of particles. Simulations using Zabrodsky's correlations predicted the rapid quenching of a hot gas flowing into a cold fluidized bed. Moreover, since the difference in temperature between the phases is not large, the error due to the use of above correlations can be neglected.

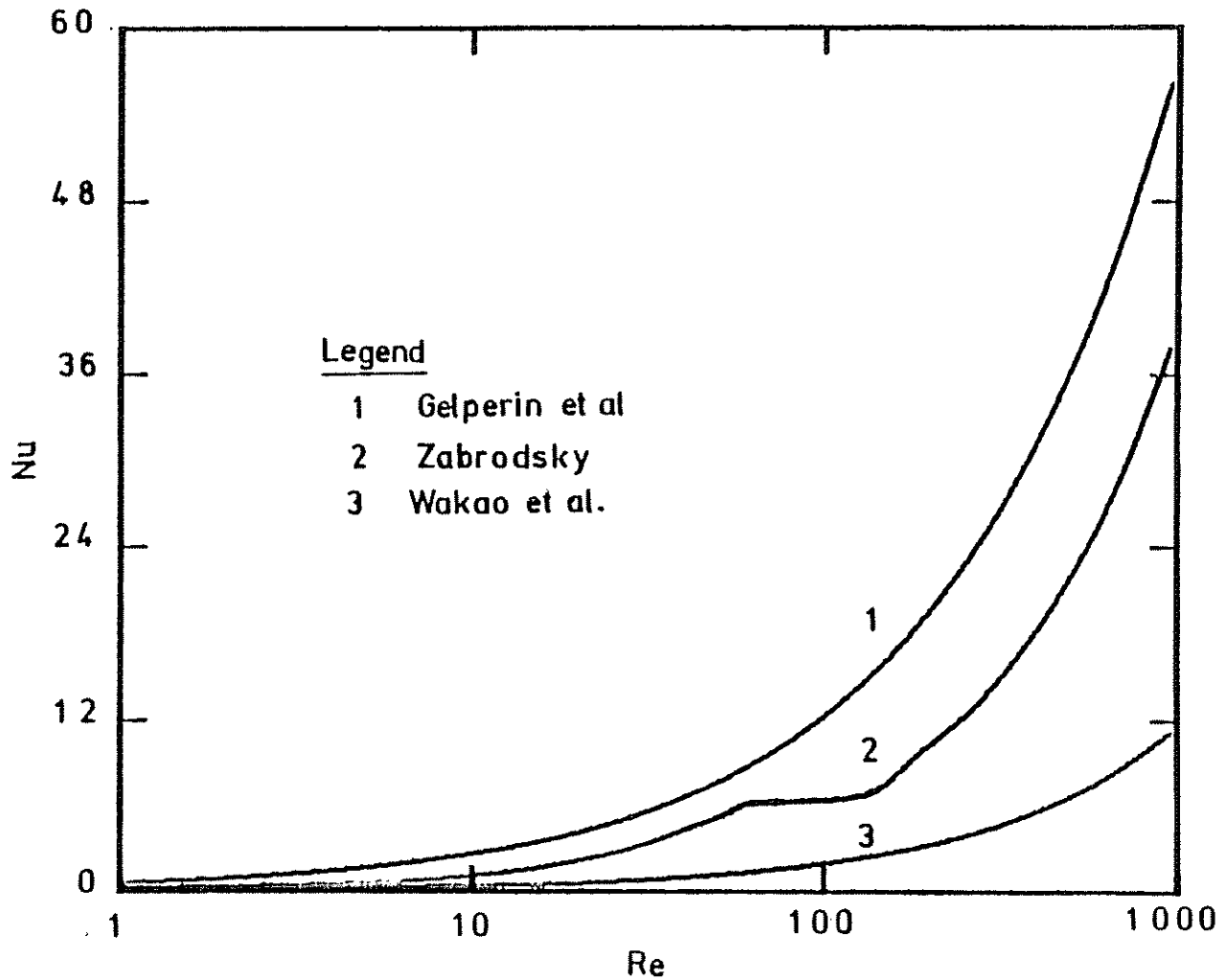


Figure 4.2. Fluid-Particle Heat Transfer

Table 4.1 Interphase Heat Transfer Correlations

If $\varepsilon \leqq 0.8$

$N_{up} = 0.106\ \bar{Re}\ S_p$ $\qquad \bar{Re} \leqq 200$

$N_{up} = 0.123\ (\frac{4\ \bar{Re}}{dp})^{0.83}\ S_p^{\ 0.17}$ $\qquad 2000 \geqq \bar{Re} > 200$

$N_{up} = 0.61 \cdot Re^{0.67} \cdot S_p$ $\qquad \bar{Re} > 2000$

If $\varepsilon > 0.8$

$N_{up} = (2 + 0.16\ \bar{Re}^{0.67})\ S_p$ $\qquad \bar{Re} \leqq 200$

$N_{up} = 8.2\ \bar{Re}^{0.6} \cdot S_p$ $\qquad 1000 \geqq \bar{Re} > 200$

$N_{up} = 1.06\ \bar{Re}^{0.457} \cdot S_p$ $\qquad \bar{Re} > 1000$

where $\bar{Re} = \dfrac{d_p\ \rho_g\ [(V_g - V_s)^2 + (U_g - U_s)^2]^{0.5}}{\mu_g}$,

$S_p = (1 - \varepsilon)\ \dfrac{6}{d_p}$

and $N_{up} = \dfrac{h_v d}{K_g}$

Particulate Phase Heat Transfer

The heat transfer between the particles poses a major difficulty in formulating the equations. In equations (4.3) it is written in the standard Fourier-law form of $\frac{\partial}{\partial x}$ (K_s (1 - ε) $\frac{\partial T}{\partial x}$) etc. Here K_s is not the conductivity of the material of the particles, but that of a collection of particles considered together as a particulate phase. There is no work known to the author dealing with the constitutive relation for K_s in a fluidized bed. In the detailed models proposed so far particle-particle interaction is usually neglected, assuming point contact between the particles. Bauer and Schlunder (1978) point out that in a packed bed point contact between the particles, which is usually assumed, does not correspond to the actual situation. In a fluidized bed the situation is slightly different due to the collisions and abrasions among the particles. It will be seen later that these phenomena appear to enhance the particulate phase heat transfer in a fluidized bed. In this study the particulate phase heat transfer is assumed to be significant.

Particle to particle transfer is a complex phenomenon, involving contact conductance, conduction through the thin layer of gas sticking to the particles, and radiation. Thus there might be some overlap of mechanisms when empirical correlations are used for calculating interphase heat transfer and particulate phase heat transfer. However, this is not a great concern here since in the near the wall region where the particulate phase heat transfer is significant, the interphase heat transfer is insignificant.

There is not a great deal of experimental evidence on the role of particulate phase heat transfer. Conclusions based on the dependence of

heat transfer coefficient on the thermal conductivity of the particulate material, K_p, might be misleading. In a model proposed by Botterill (1975), there is a strong dependence of the heat transfer coefficient on the thermal conductivity of the particulate material. Zabrodsky et al. (1978), however, cite experimental data in which fluidization using glass beads and copper shots gave similar heat transfer coefficients. Thus they conclude that at large residence times of particles at the heat transfer surface the influence of K_p is negligible. Though the above assertion may be partly true, the particulate phase heat transfer is by no means insignificant. Wen and Chang, as cited by Botterill (1975), estimated that 10 to 35% of the total heat was transferred by particle/particle transfer. Another way of analyzing this mode of transport might be by looking at the effective conductivities of packed beds, as shown in Figure 4.3. There a correlation due to Baskakov (Gelperin and Einstein 1971) for effective bed conductivity is shown together with the two extreme conditions: $K_s = K_p$ and $K_s = 0$. The fallacy of "$K_s = 0$" is obvious. Also, shown is a theoretical prediction due to Maxwell (Parrott and Stuckes, 1975), where he considers the dispersed phase as being covered by a film of continuous phase to get the effective conductivity of the mixture as:

$$\frac{K_{bed}}{K_g} = \frac{1 + 2K_g/K_s - 2\,(1 - \varepsilon)\,(K_g/K_s - 1)}{1 + K_g/K_s + (1 - \varepsilon)\,(K_g/K_s - 1)} \qquad (4.4)$$

Even this model predicts a conductivity smaller than that of Baskakov's. Hence, we conclude that it is essential to consider particulate phase heat transfer and that this mode of transport takes place parallel to the gas phase heat transfer.

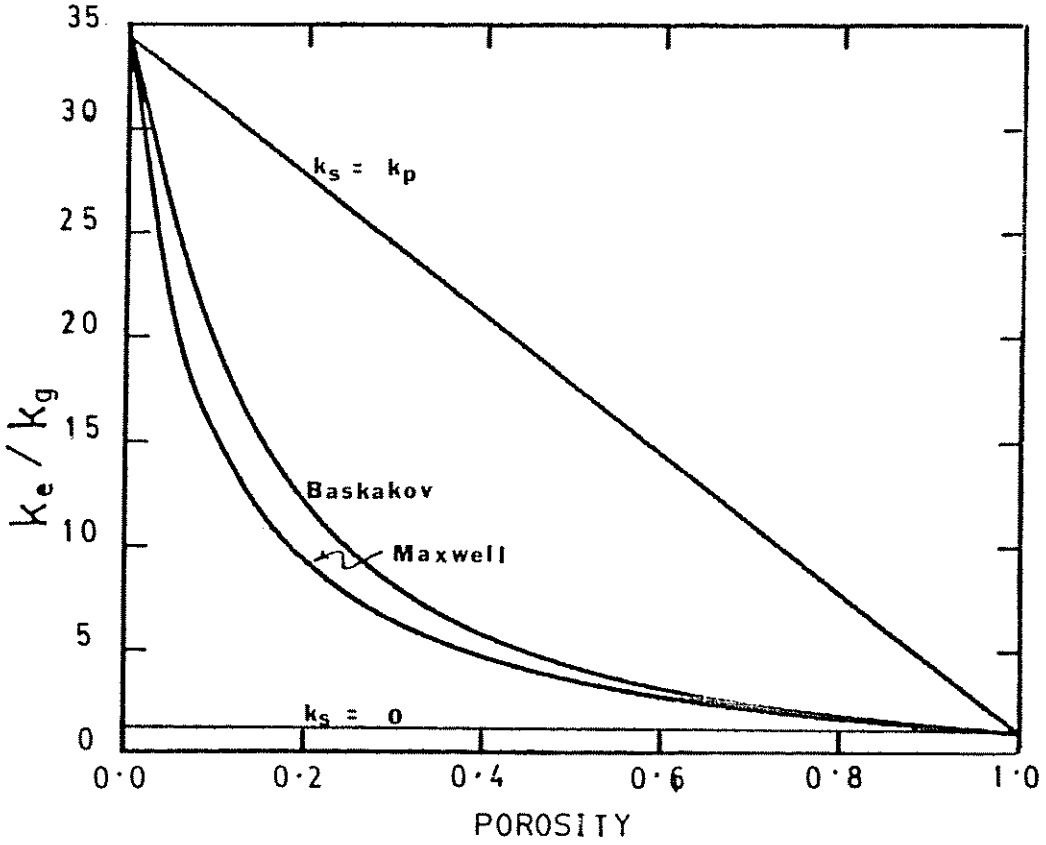


Figure 4.3. Effective Bed Conductivity as a Function of Porosity

A good approximation for the particulate phase conductivity is the effective radial thermal conductivity of solids in a packed bed. Nayak and Tien (1978) used a statistical thermodynamic approach to calculate the coordination number distributions and subsequently to calculate the lattice conductivity using a Holm formula. This appears to be a promising approach for modeling fluidized bed particulate heat transfer. However, such an extension is not being attempted here due to the lack of sufficient experimental data to test the model. Here we use a model due to Zehner and Schlunder (Bauer and Schlunder, 1978) for the effective radial conductivity of packings. Neglecting the radiation and Smoluchowski effects, the equations reduce to the following:

$$\frac{K_s}{K_g} = (1 - \sqrt{1 - \varepsilon}) + \sqrt{1 - \varepsilon}\ [\phi R + (1 - \phi) \frac{\lambda^*_{so}}{\lambda}] \qquad (4.5)$$

where

$$\frac{\lambda^*_{so}}{\lambda} = \frac{2}{(1 - B/R)} [\frac{\frac{B}{R}(R - 1)}{(1 - B/R)^2} \ln \frac{R}{B} - \frac{B - 1}{(1 - \frac{B}{R})} - \frac{B + 1}{2}]$$

$$R = \frac{K_p}{K_g}$$

$$B = 1.25 \left(\frac{1 - \varepsilon}{\varepsilon}\right)^{10/9} \quad \text{for spheres}$$

$$\phi = 7.26\ 10^{-3}$$

For glass/air system the above formula is shown in Figure 4.4. Note that in the operating region of fluidized beds ($\varepsilon > 0.4$), K_s is significantly smaller than K_p and is more comparable to K_g. This might

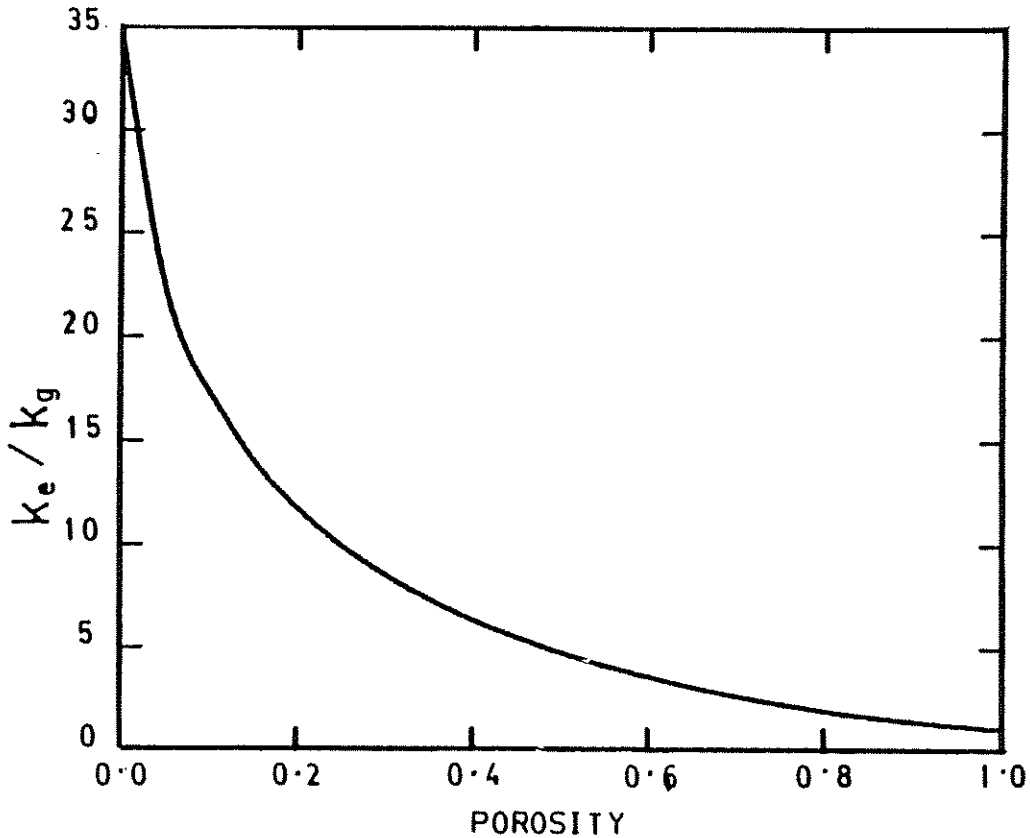


Figure 4.4. Zehna and Schlunder Model for Effective Radial Conductivity of Solids in a Packed Bed

be the reason for the apparent absence of the influence of K_p on the heat transfer coefficient as reported by Zabrodsky et al. (1978).

In some of the detailed models it is customary to assume a gap of "0.1 dp" between the particles and the wall. Though, there is an increase in the bed porosity near the wall (Ozkaynak and Chen, 1980; Pillai, 1977), it is not realistic to assume such a gap in a continuum model. For an average bed porosity of 0.45, Gelperin and Einstein (1971) use a wall porosity of 0.476. Such a wall effect is neglected in this study.

Numerical Solution

The set of nonlinear partial differential equations is solved for U_s, U_g, V_s, V_g, ε, T_g, T_s and P using the ICE method (Rivard and Torrey, 1977; Ettehadieh, 1982). Velocities of each phase are centered on the cell boundaries. Quantities such as density, porosity temperature and pressure are centered at the center of the mesh. Porosities at other locations are found by linear interpolation. Flux terms in continuity and momentum equations are full donor-cell differenced. Convective terms in the momentum equations are advanced in time explicitly, whereas all exchanges of mass, momentum and energy, when they occur, are treated implicitly. The resulting finite difference equations are solved by a combination of point relaxation, Newton's and secant iteration method, without any linearization.

Due to the vigorous circulation of the solids, the bulk of the fluidized bed is practically isothermal. Trial simulations proved that the significant mechanism in the bed to wall heat transfer is a transient transport phenomenon very close to the wall. This conforms to the "packet model" of Mickley and Fairbanks in which they consider the

mechanism to be transient conduction to a packet of particles which resides at the heat transfer surface for a short while. Also there is no evidence of a steady-state film of gas or solids near the wall offering a resistance to heat transfer (Saxena et al., 1978). For "capturing" such a transient phenomenon we need much finer computational cells near the wall than those used for the hydrodynamic simulation. The hydrodynamic equations were solved using coarse grids to minimize computational time. The energy equations were also solved over this computational grid except for a column of cells near the wall. These were divided into subcells for better resolution. The velocities required in these subcells were computed assuming a linear variation. The number of subcells was increased till convergence was obtained, as shown in Figure 4.5.

The primary goal in this study has been to use the continuum approach to model fluidized bed heat transfer and to establish that the enhancement of the heat transfer coefficient is predictable using simple mechanisms. In addition, we compare our results with the experimental data of Ozkaynak and Chen (1980) for a vertical tube immersed in a uniformly fluidized bed. In a uniformly gas-fluidized bed bubbles form at random over the distributor plate. However, there is a preferential formation of bubbles near the wall, due to the higher porosity in that region. In the computer simulation, we use a non-uniform gas flow rate at the distributor plate; the gas flow rate near the wall is higher compared to the rest of the bed. However, it was found that the average heat transfer coefficient is insensitive to any particular gas flow rate distribution and that it is sensitive only to the total gas flow rate.

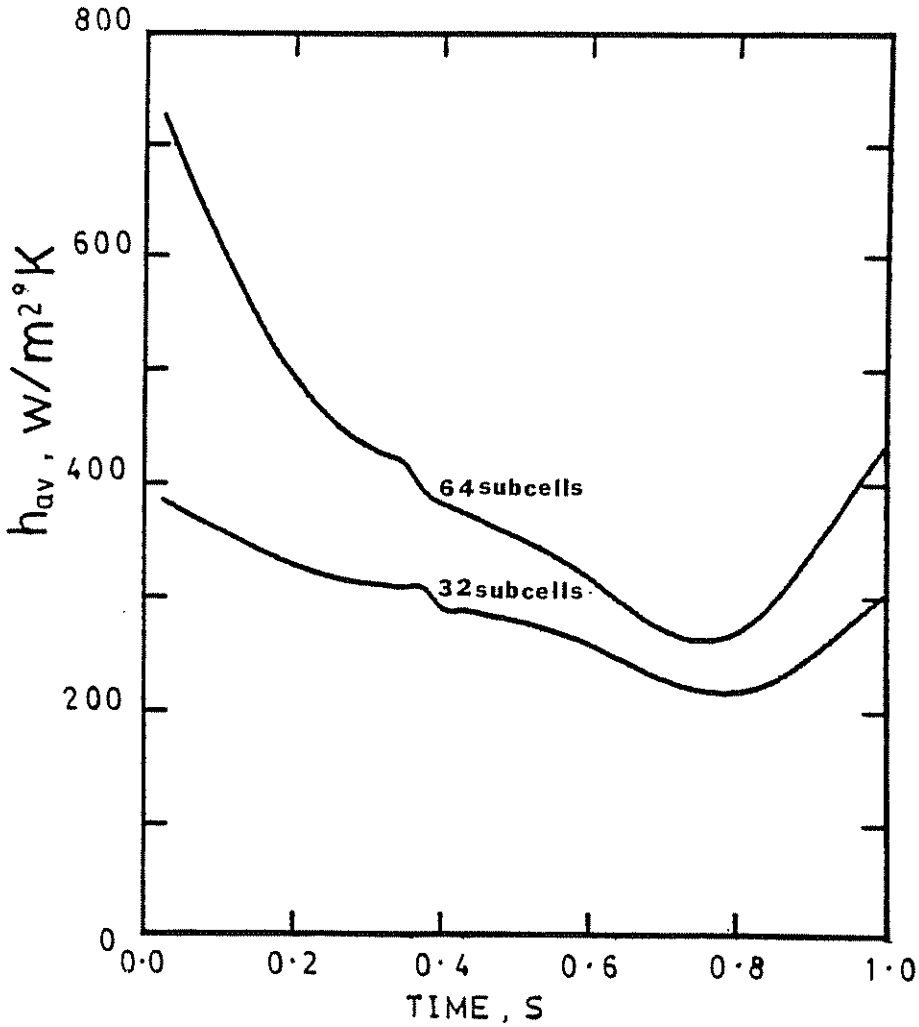


Figure 4.5. Convergence as the Number of Subcells is Increased

A summary of the simulation conditions used to simulate Ozkaynak and Chen's experiment is shown in Table 4.2.

Table 4.2 Conditions for the Heat Transfer Simulation

Particle diameter (μm)	610
Particle thermal (W/m°K) conductivity	0.89
Particle heat capacity ($J/K_g{}^oK$)	753.6
Particle density (K_g/M^3)	2.470

Results and Discussions

In Figure 4.6 transient bed averaged heat transfer coefficients for various gas flow rates are shown. The bed averaged wall-to-bed heat transfer coefficient is defined as:

$$h_{av} = \frac{q_{av}}{(T_w - T)} \tag{4.6}$$

The transient behavior is similar in all the cases. Initially there is a rapid decay in the value of heat transfer coefficient as a temperature profile develops into a nearly static bed. But there are two more phenomena taking place simultaneously in the bed: 1) bed circulation, which disrupts the development of the temperature profile

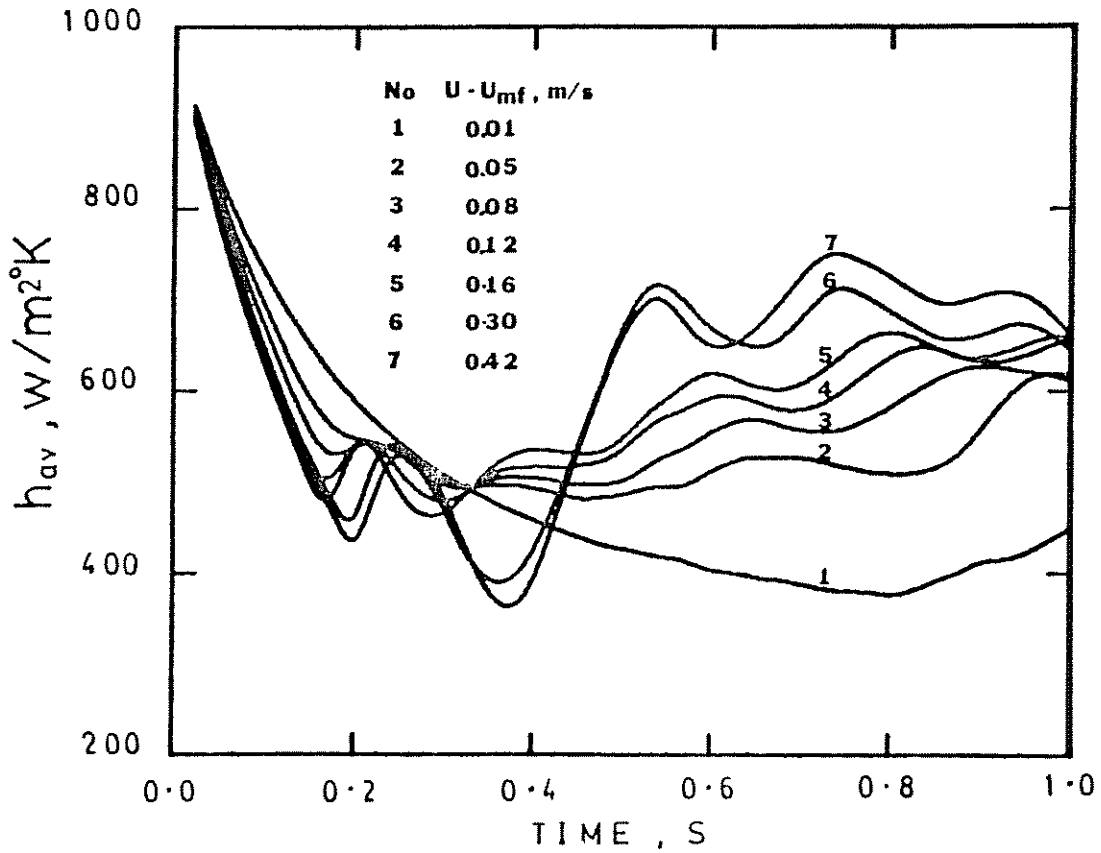


Figure 4.6. Average Bed-to-Wall Heat Transfer Coefficients for Different Gas Flow Rates

and thus tends to increase the heat transfer coefficient, 2) growth and propagation of the start-up bubble, which tends to decrease the heat transfer coefficient. The effect of above three mechanisms is more pronounced at higher gas flow rates. After 0.5s the heat transfer coefficients reach a nearly oscillatory steady state. Hence, we compute the time averaged heat transfer coefficients by integrating the instantaneous heat transfer coefficients from 0.5 to one second.

In Figure 4.7 the average heat transfer coefficients are plotted as a function of $U-U_{mf}$, together with experimental data of Ozkaynak and Chen and the predictions of a correlation due to Wender and Cooper (Ozkaynak and Chen, 1980). Curve 1, shows the theoretical predictions when $K_s = K_p$. As expected, the predictions are much higher than the experimental values. However, the theory correctly predicts the sharp rise in the heat transfer coefficients in the beginning, and the leveling off thereafter. Curve 2 shows the theoretical predictions when the Zehner and Schlunder model for K_s was used. Now the theoretical predictions are reasonably close to the experimental data. It may be that the actual K_s should be higher than that predicted by Zehner and Schlunder model. It must be emphasized that no fitting parameters have been used in the computer simulations. It can be seen that the prediction of the continuum model is superior to the predictions of the correlation.

In Figure 4.8, the propagation of thermal waves into the bed is shown for different time intervals after startup in terms of the gas temperature profiles. Ozkaynak and Chen (1980) observe that the thermal waves penetrate up to a depth of one particle diameter from the wall. In Figure 4.8a, b we observe the same phenomena. It is seen that as

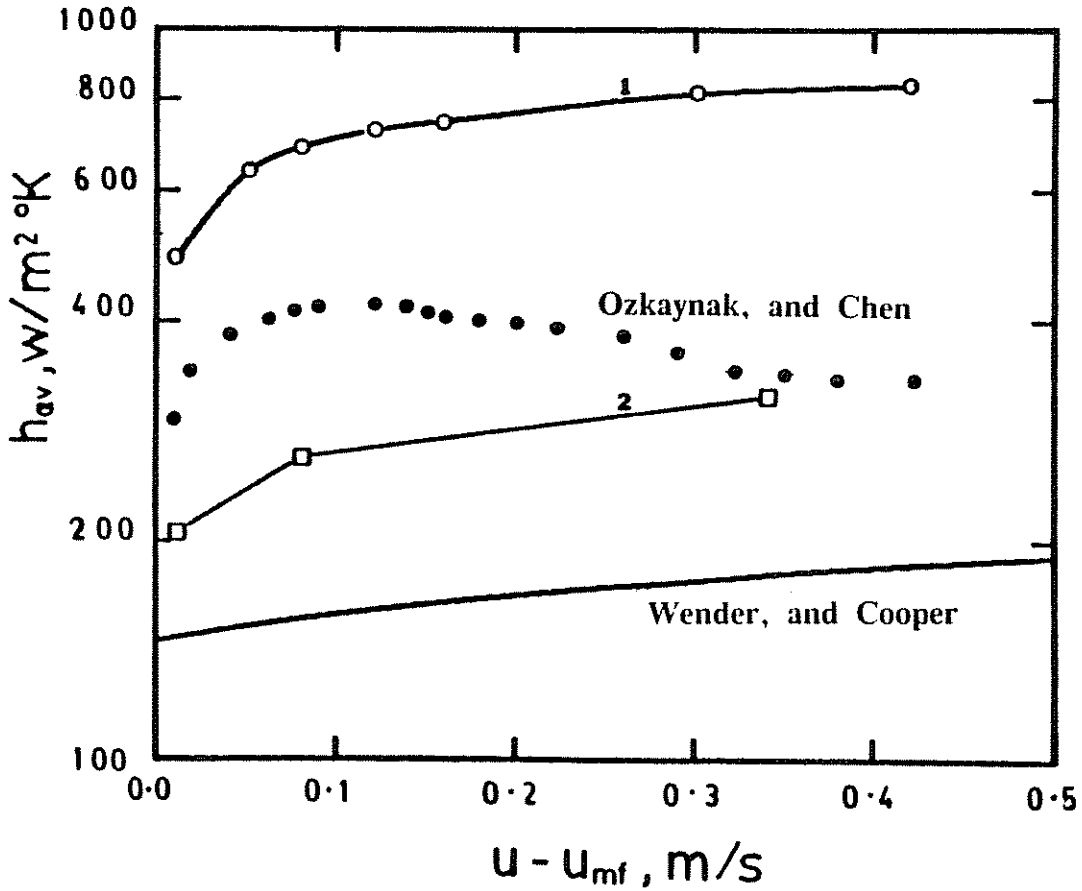


Figure 4.7. Comparison of Average Heat Transfer Coefficients With Experimental Data

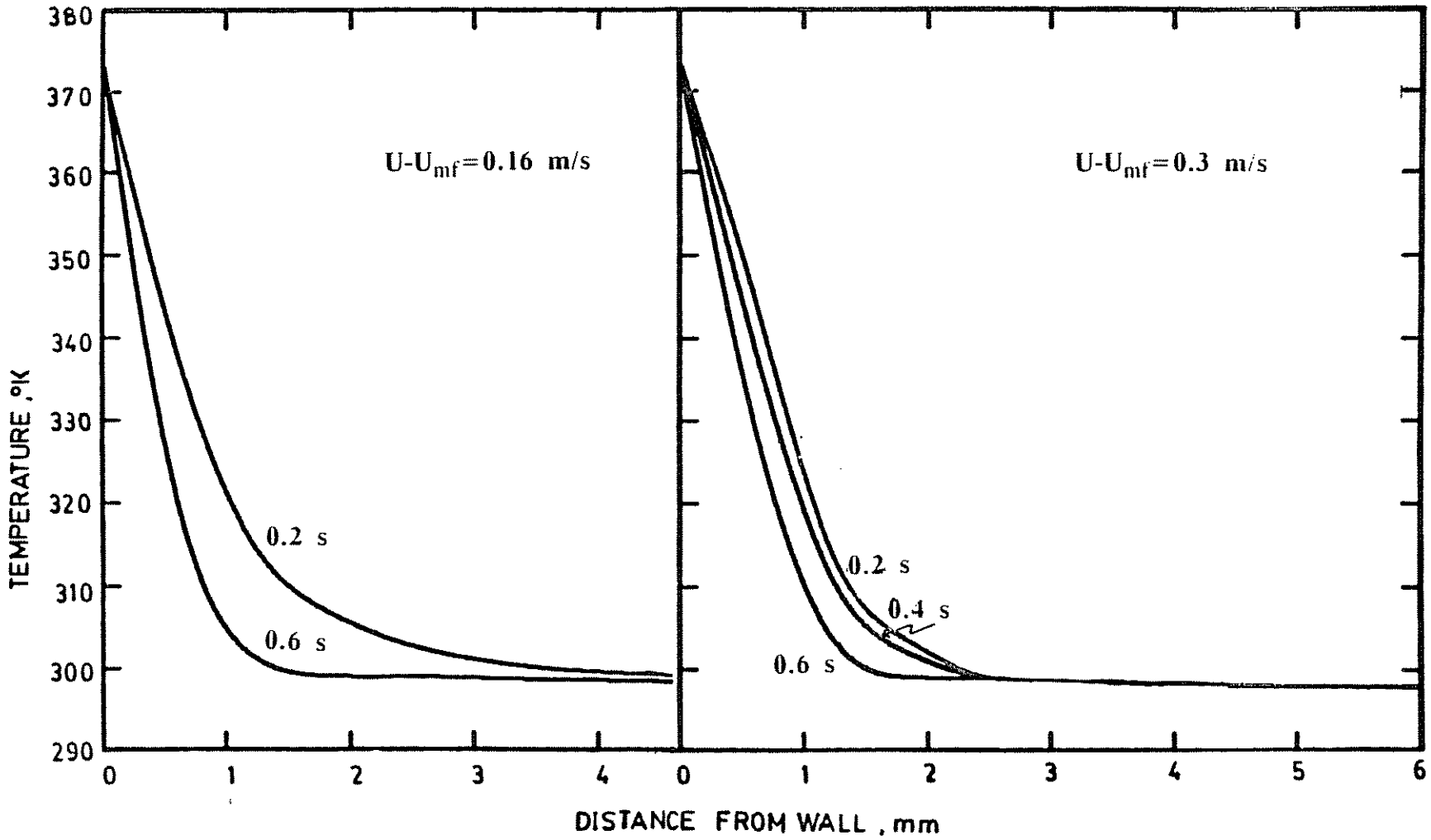


Figure 4.8. Gas Temperature Profiles Near the Wall

time passes the temperature profile becomes steeper. After about 0.4s the change in the temperature profile is not much; i.e. some kind of a "thermal equilibrium" has been reached near the wall. It is also seen that the behavior of the thermal waves at different gas flow rates is similar.

In Figure 4.9 the instantaneous local heat transfer coefficients at two different locations in the bed are shown. The effect of the startup transients is visible at both the locations. At the higher location, which is nearly at the top of the bed, the startup transients may be seen to leave the bed at about 0.5s. After this there are nearly regular fluctuations in the local heat transfer coefficients. In Figure 4.10, the porosity, axial gas velocity, and axial solids velocity at the lower location are shown. Qualitatively we can correlate the local heat transfer fluctuation to the local porosity fluctuation. To some extent there are some fluctuations in the local gas velocity also, whereas the local solids velocity remains steady. However, from the above figures it is clear that the behavior of the heat transfer coefficient is not amenable to a simple qualitative treatment.

One of the useful aspects of the continuum approach is its ability to provide detailed information on the local behavior of the variables. Figures 4.8 and 4.10 are examples. Further, in Figure 4.11 the contribution to the heat transfer coefficient from the particulate phase and gas phase heat transfer are shown. Such quantities are not easy to measure.

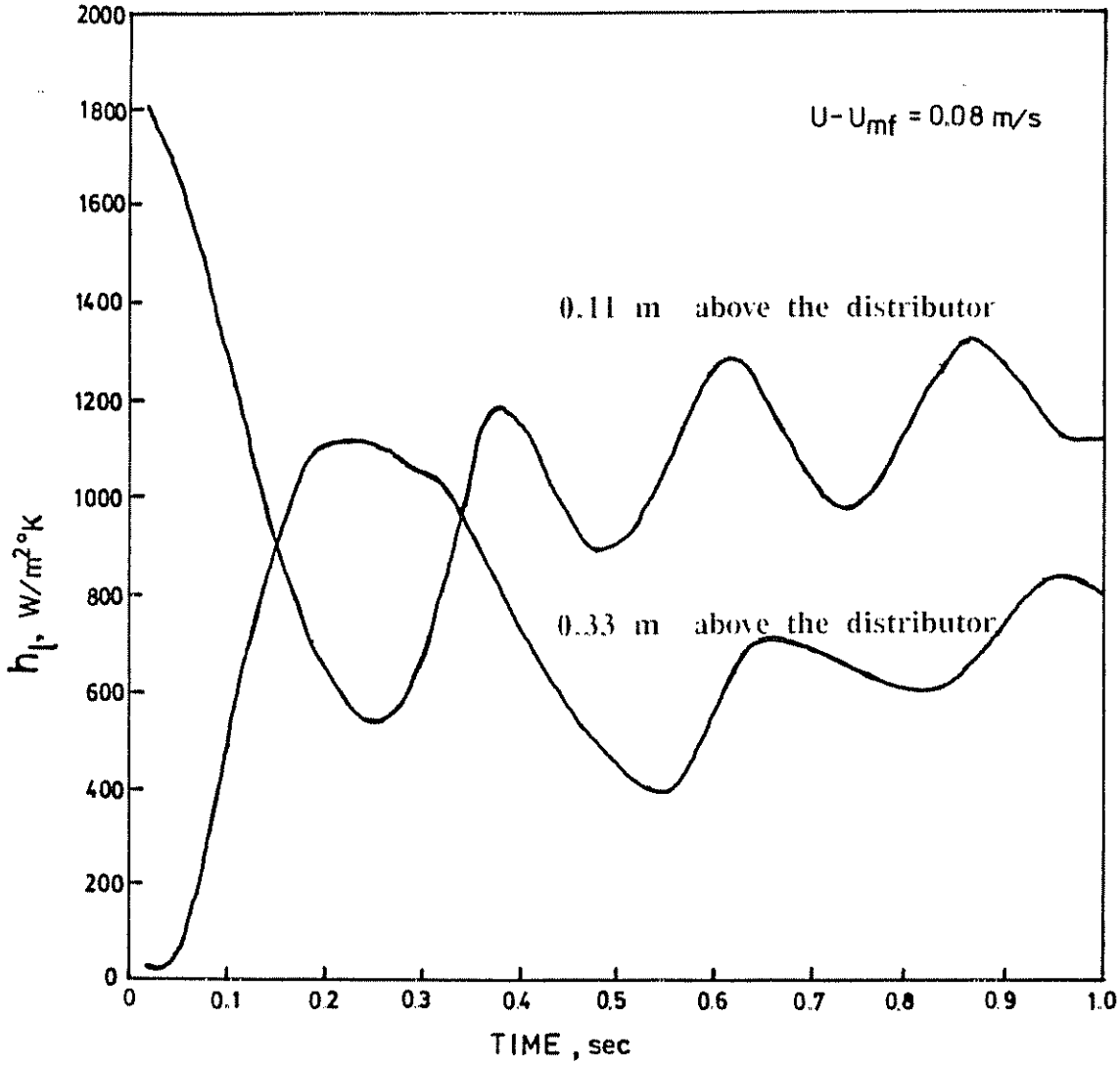


Figure 4.9. Local Heat Transfer Coefficients at Two Locations

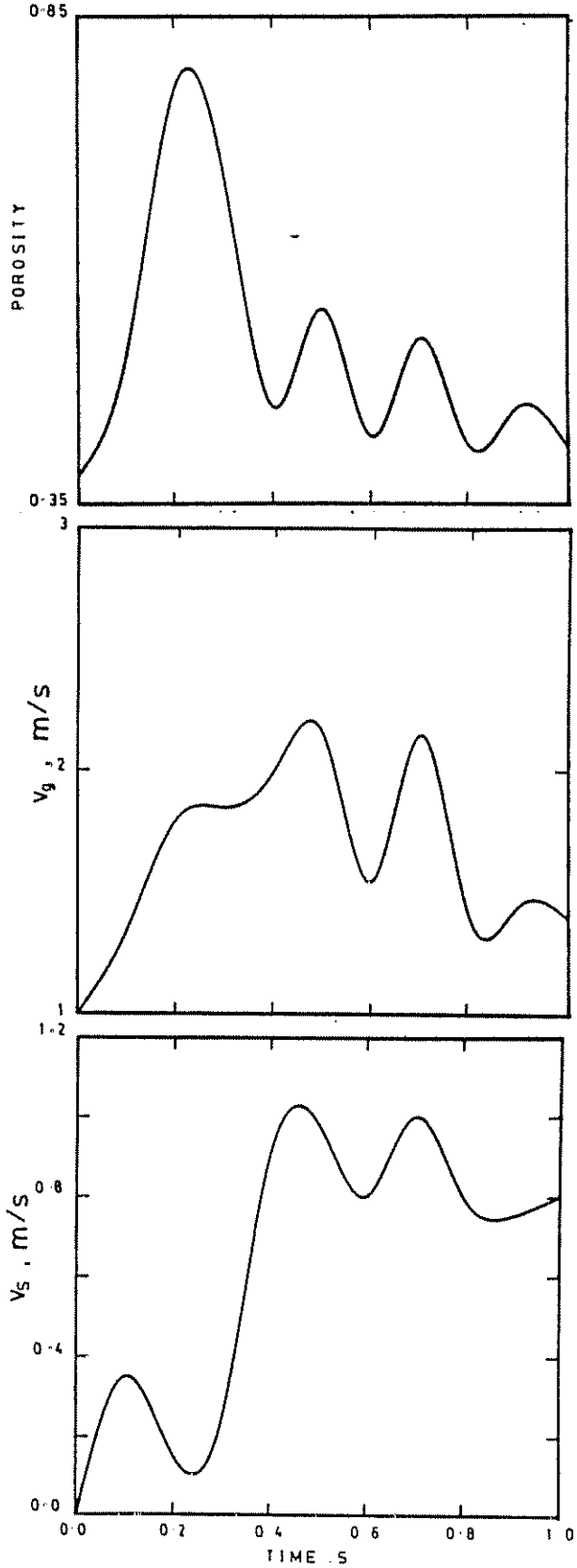


Figure 4.10. ε, V_g, and V_s at 0.11m Above the Distributor ($U - U_{mf}$ = 0.16 m/s)

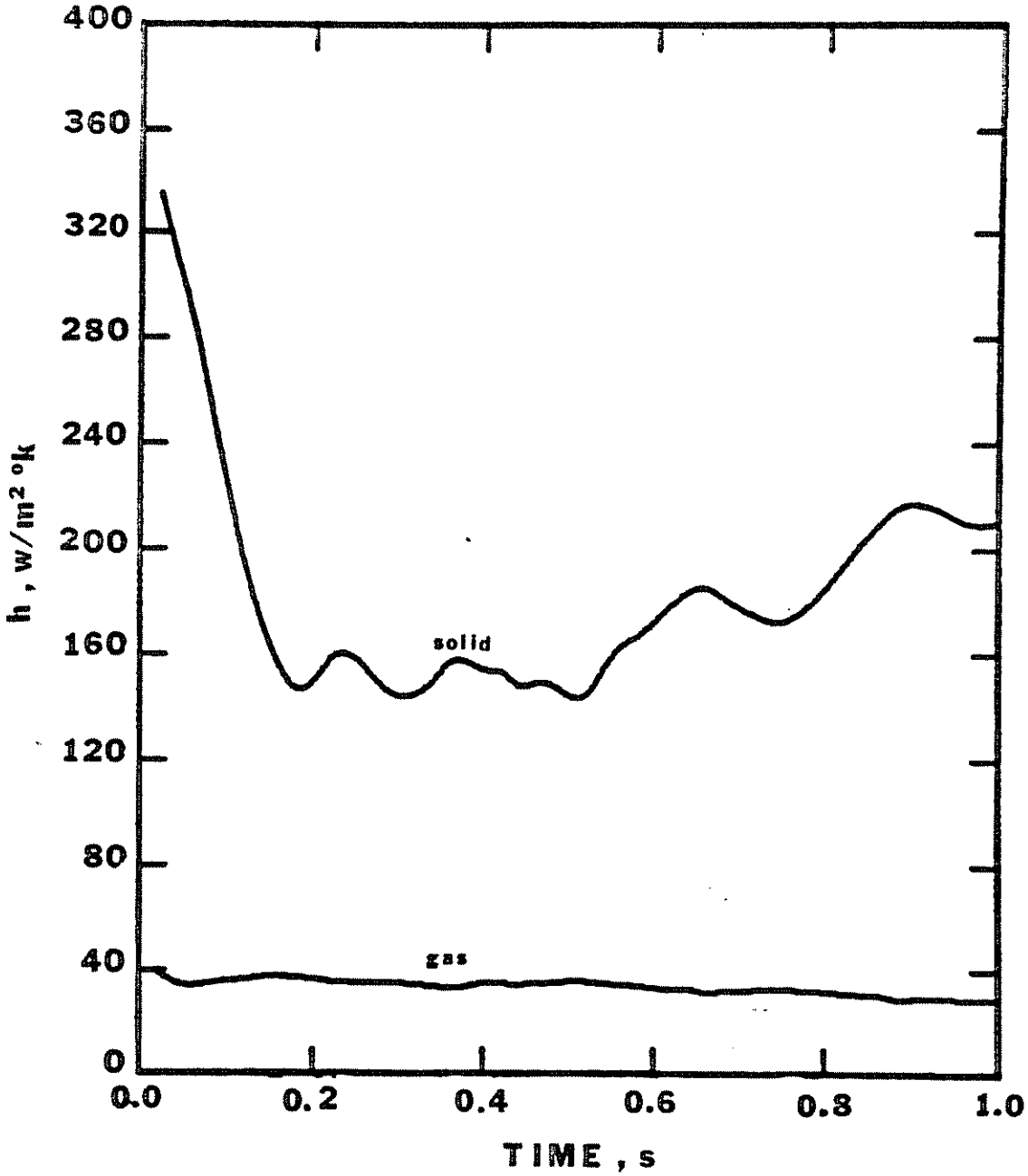


Figure 4.11. Contributions to Heat Transfer Due to Gas Phase and Particulate Phase Conduction

CHAPTER V

DILUTE PHASE TRANSPORT

Radial Velocity Profiles

The computations presented in the previous chapters were for the dynamics in the dense beds. The same equations may be used for simulating dilute phase transport such as in pneumatic conveyors. Here we present two studies involving a dilute transport system. For the first case we modify the equations developed in Chapter II by introducing viscosity terms in the momentum equations as shown below:

$$\frac{\partial}{\partial t}(\rho_k \varepsilon_k) + \nabla \cdot (\varepsilon_k \rho_k V_k) = 0 \tag{5.1}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_k \varepsilon_k V_k) + \nabla \cdot (\varepsilon_k \rho_k V_k V_k) = &- \varepsilon_k \nabla P + \rho_k \varepsilon_k g \\ &+ G_K \nabla \varepsilon_1 + \sum_{\ell=1}^{N} K_{k\ell}(V_\ell - V_k) \\ &+ \nabla \cdot \varepsilon_k \mu_k \nabla V_k \end{aligned} \tag{5.2}$$

where

$$\sum_{k=1}^{N} \varepsilon_k = 1 \tag{5.3}$$

The constitutive relations are given in Chapter II.

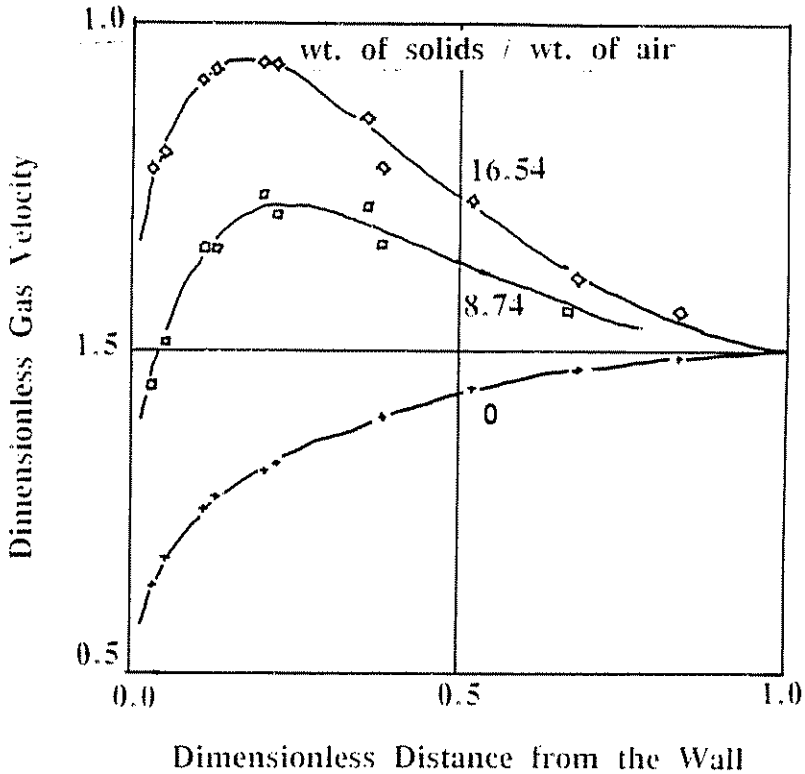
Vertical pneumatic conveying under the conditions shown in Table 5.1 was modeled using the above equations in two dimensions for a single particle size.

Table 5.1 Conditions for the Pneumatic Conveyor Simulation

Particle diameter	1mm
Pipe diameter	10cm
Height of the conveyor	4m
Solids loading $\left(\frac{\text{Mass flowrate of solids}}{\text{Mass flowrate of air}}\right)$	8.74, 16.54
Air velocity at the inlet	15m/s
Void fraction at the inlet	0.996, 0.992
Solids velocity at the inlet	12.8, 13.5 m/s

These conditions are similar to those used by Vollheim (1963). A set of velocity profiles given by him has been reproduced in Figure 5.1. In this figure we find that the gas velocity shows a maximum away from the axis of the pipe. Another surprising feature is that even in the absence of solids the gas velocity profile is nearly parabolic.

Figure 5.2 shows the computed gas velocity profiles at four axial locations along the pipe. A value of 1.0 poise was used as the solids viscosity. The profile is nearly flat except for a short region near the wall. This is clearly at variance with Vollheim's data. Figure 5.3 shows the corresponding solids velocity profiles. These profiles are similar to the gas velocity profiles with an average slip of 3m/s. Figure 5.4 shows the solids volume fraction profiles. At the entrance



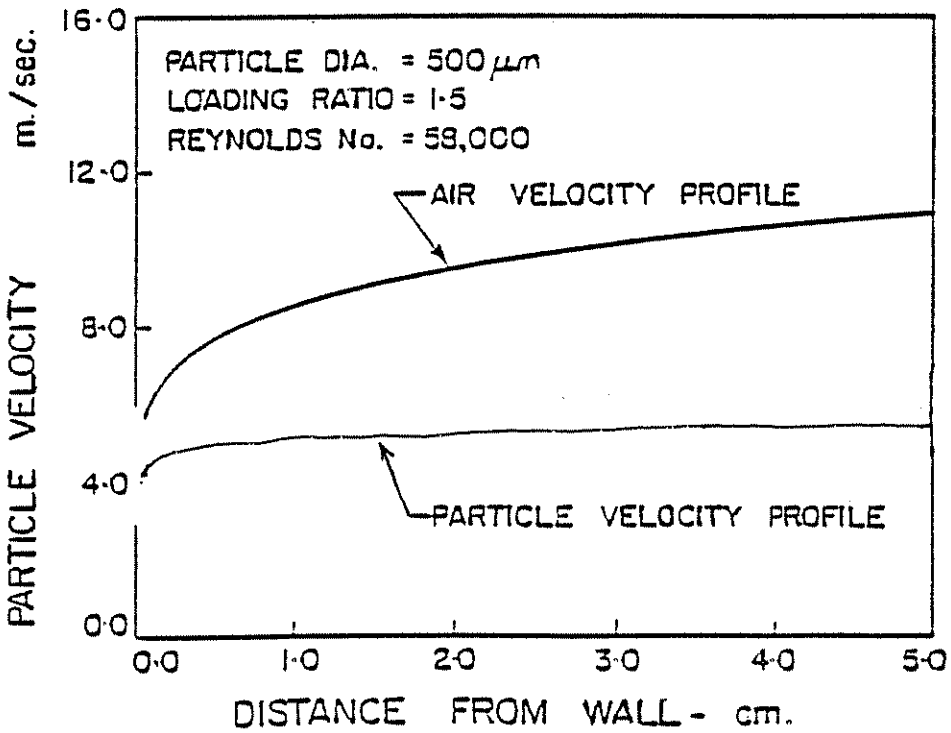


Figure 5.1. Experimental Axial Velocity Profiles

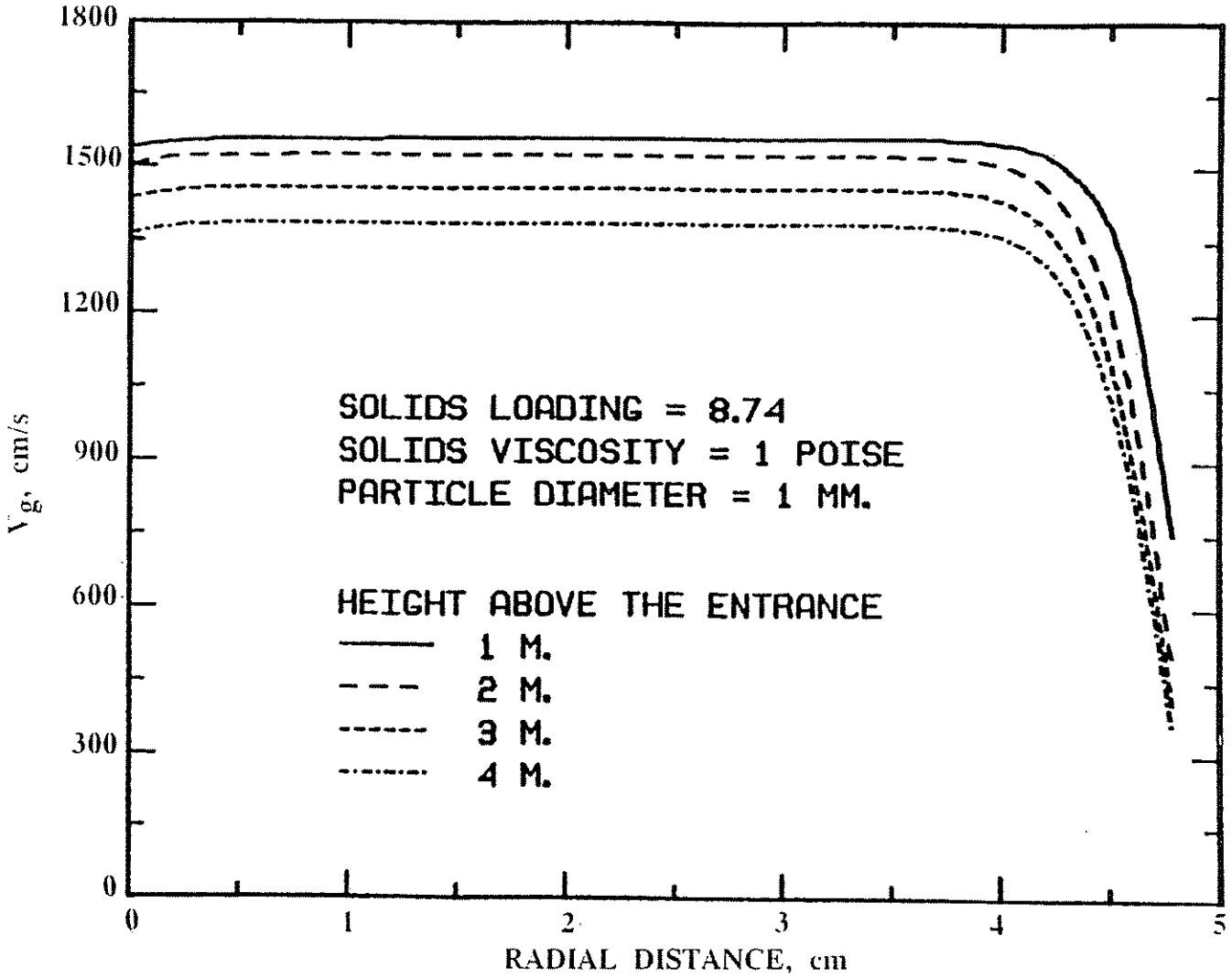


Figure 5.2. Axial Gas Velocity Profiles for μ_s = 1 Poise

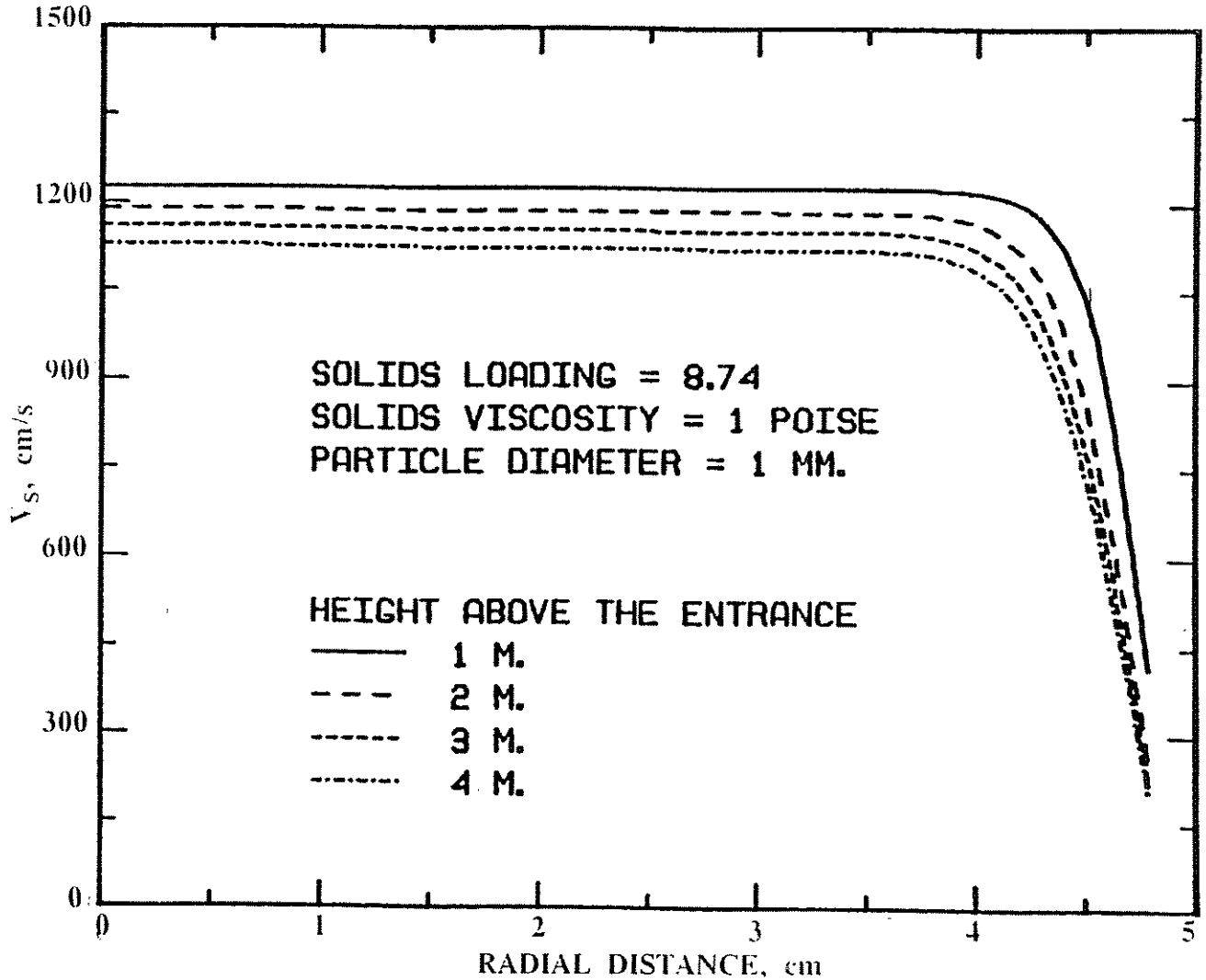


Figure 5.3. Axial Solids Velocity Profiles for μ_S = 1 Poise

the solids tend to accumulate near the wall. However, further away from the entrance the distribution is nearly uniform.

To study the effect of solids viscosity, computations were repeated using a value of 10.0 poise as the solids viscosity. Figure 5.5 shows the computed gas velocity profiles. As expected,the profiles are not flat anymore. However, these profiles are still not similar to those of Vollheim (1963). Figure 5.6 shows the corresponding solids velocity profiles. The influence of viscosity is similar to that on the gas velocity. But the magnitude of the solids velocity at the center is not affected by the solids viscosity as in the case of gas velocity. Thus, as the solids viscosity is increased the slip increases to about 5 m/s. Figure 5.7 shows a slight increase in the solids volume fraction. This compensates for the reduction in the average solids velocity as evident from Figure 5.8.

Computations were also carried out at a higher solids loading. Vollheim's data show that the maxima of the gas velocity profiles are more pronounced at higher solids loading. Computed gas velocity profiles at a higher solids loading, as shown in Figure 5.9 does not have such a behavior. The solids velocity profiles, as shown in Figure 5.10 are also similar to those in the previous cases. As expected, Figure 5.11 shows an increase in the solids volume fraction.

Figure 5.12 shows the rapid development of the axial gas velocity when a concentrated mixture (porosity = 0.8) is fed at the inlet. Figure 5.13 shows the corresponding change in the solids velocity. These figures show a relatively rapid initial change followed by a smooth change to attain a state of constant slip between the phases.

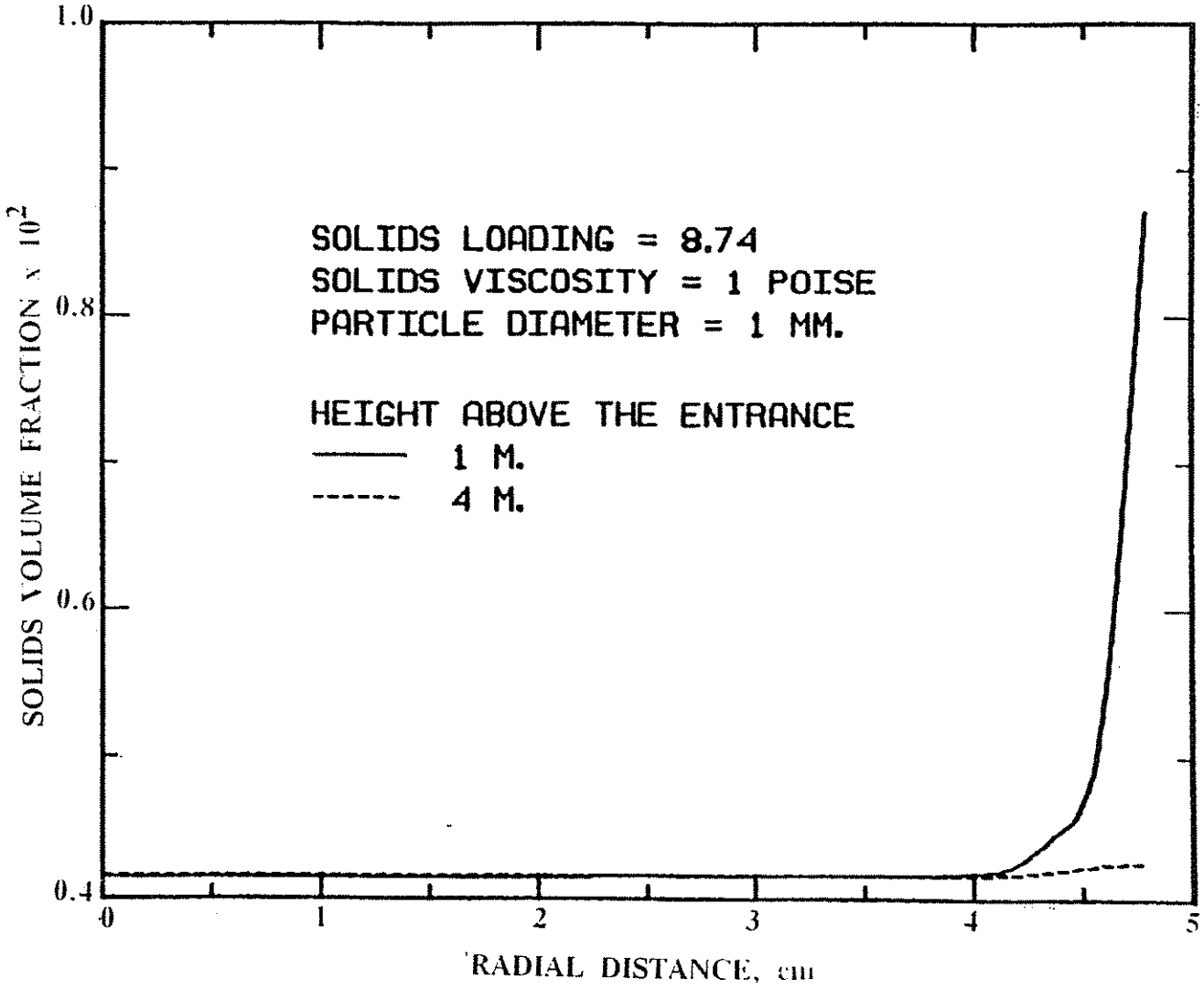


Figure 5.4. Solids Volume Fraction Profiles for μ_s = 1 Poise

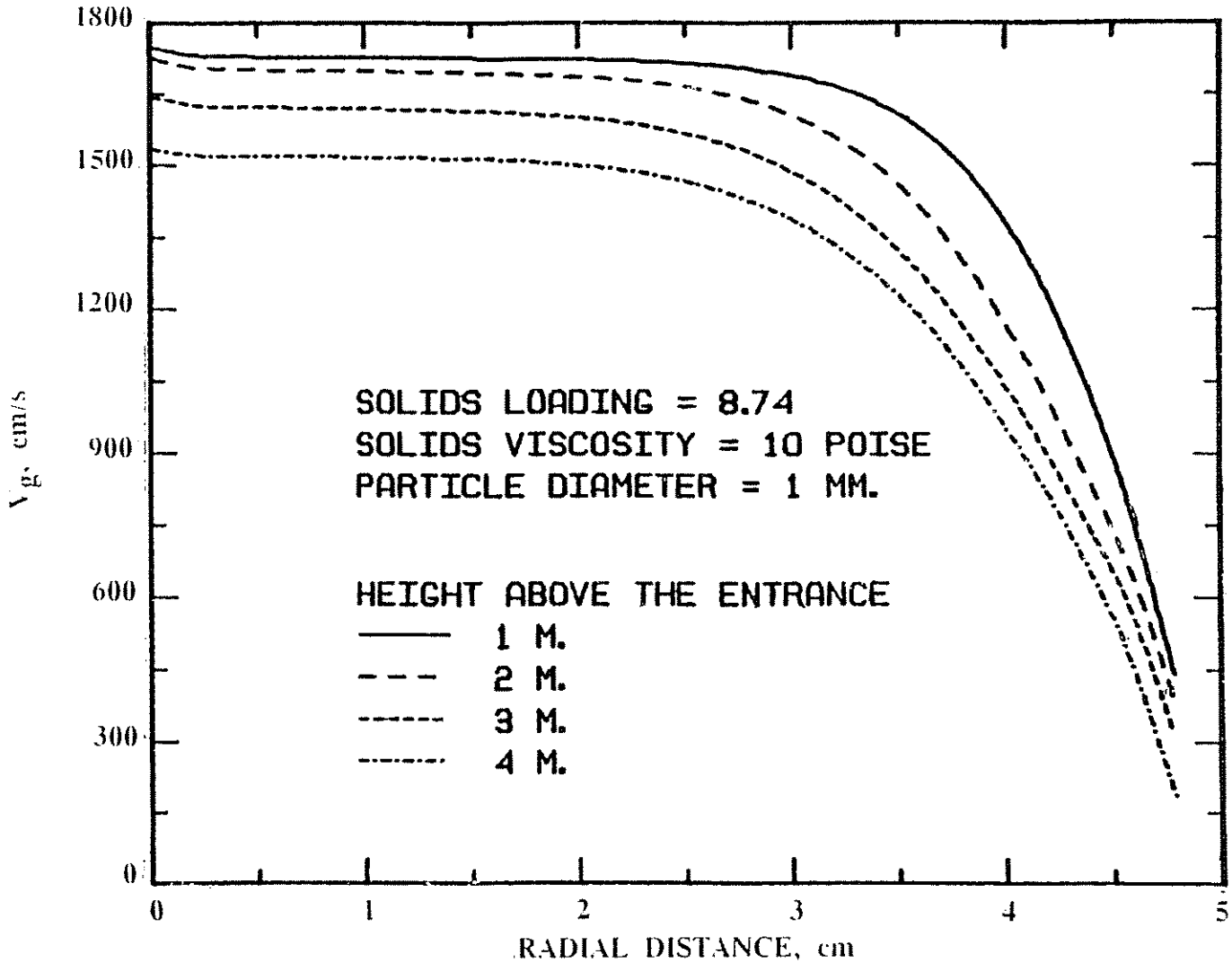


Figure 5.5. Axial Gas Velocity Profiles for μ_s = 10 Poise

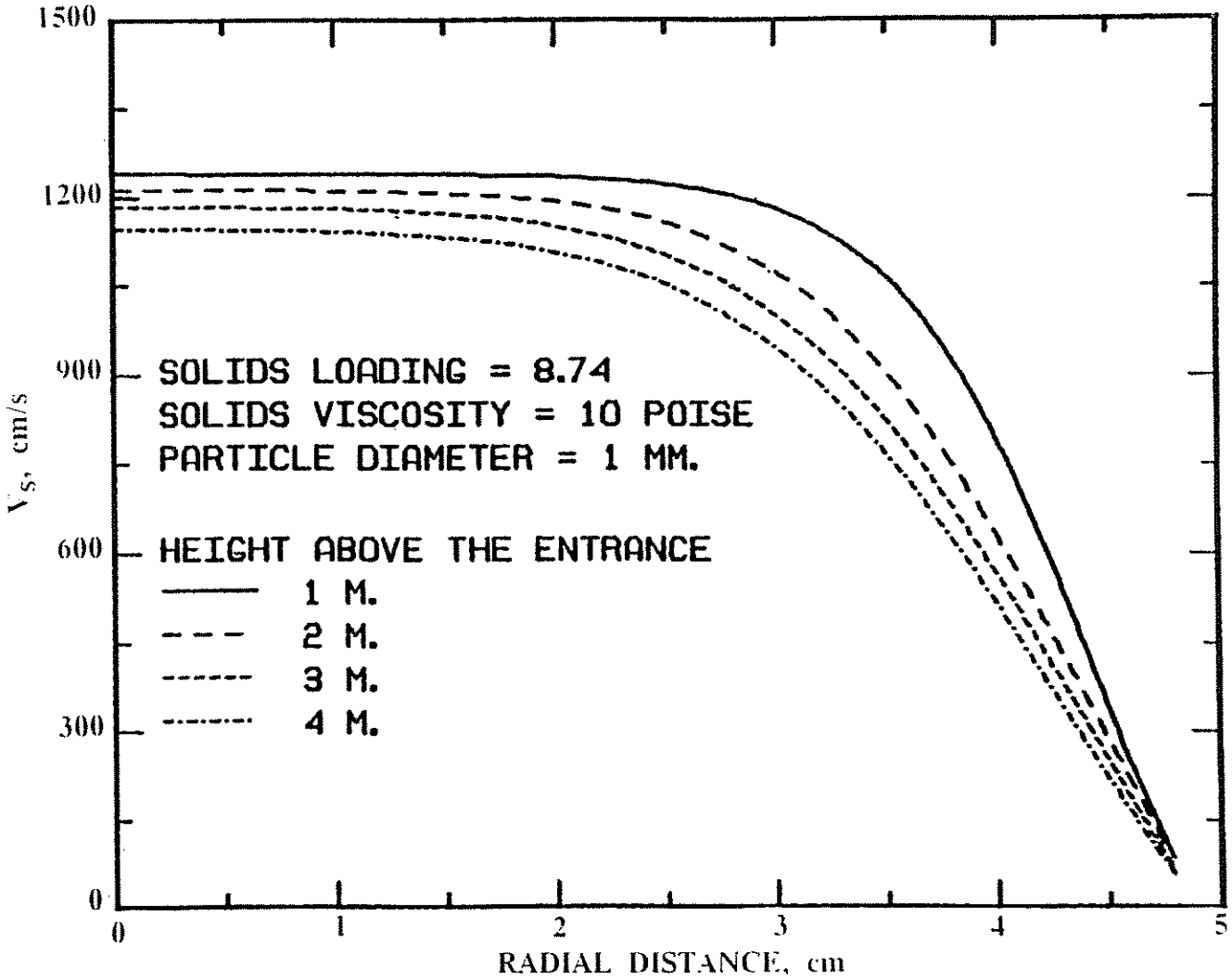


Figure 5.6. Axial Solids Velocity Profiles for μ_S = 10 Poise

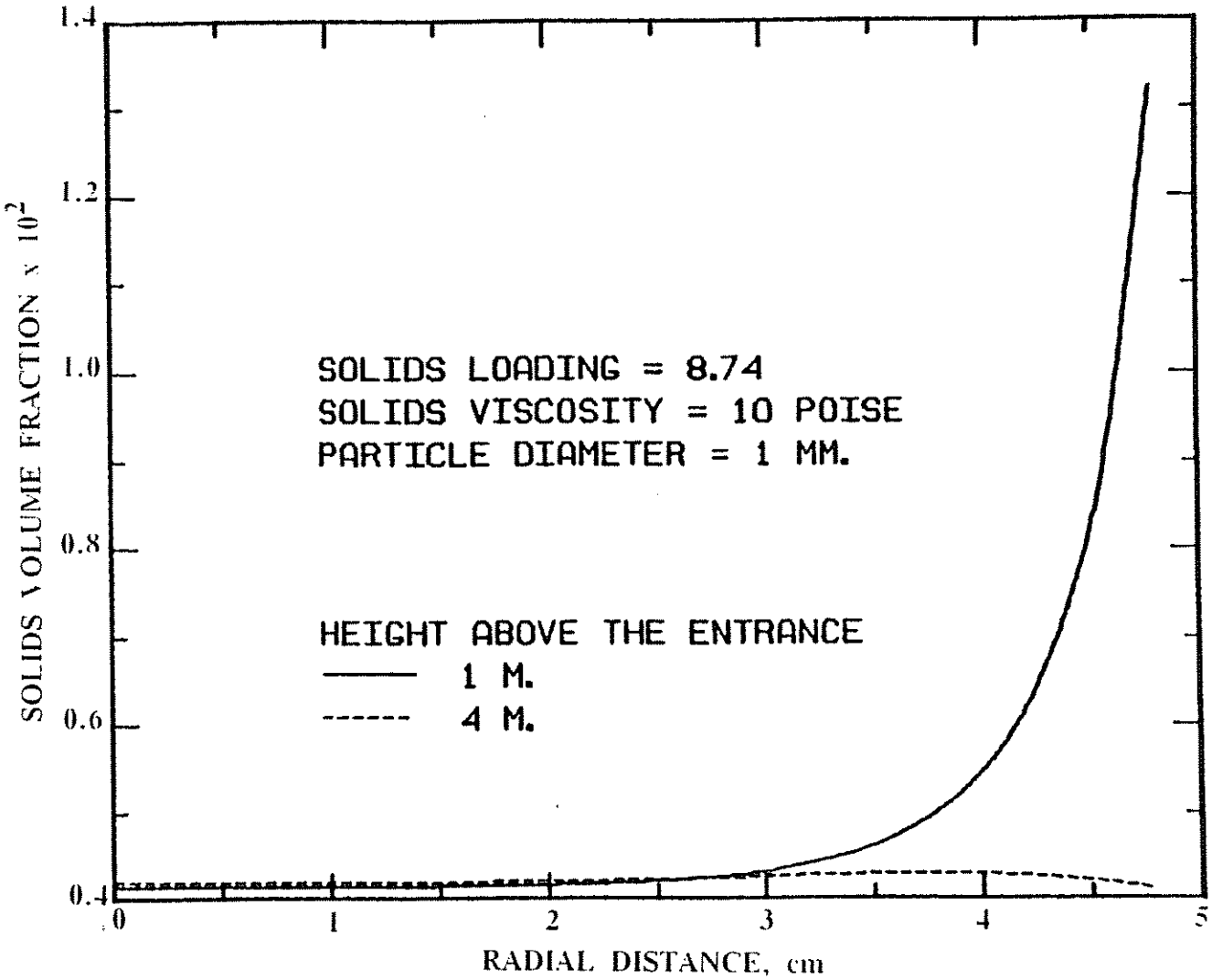


Figure 5.7. Solids Volume Fraction Profiles for μ_s = 10 Poise

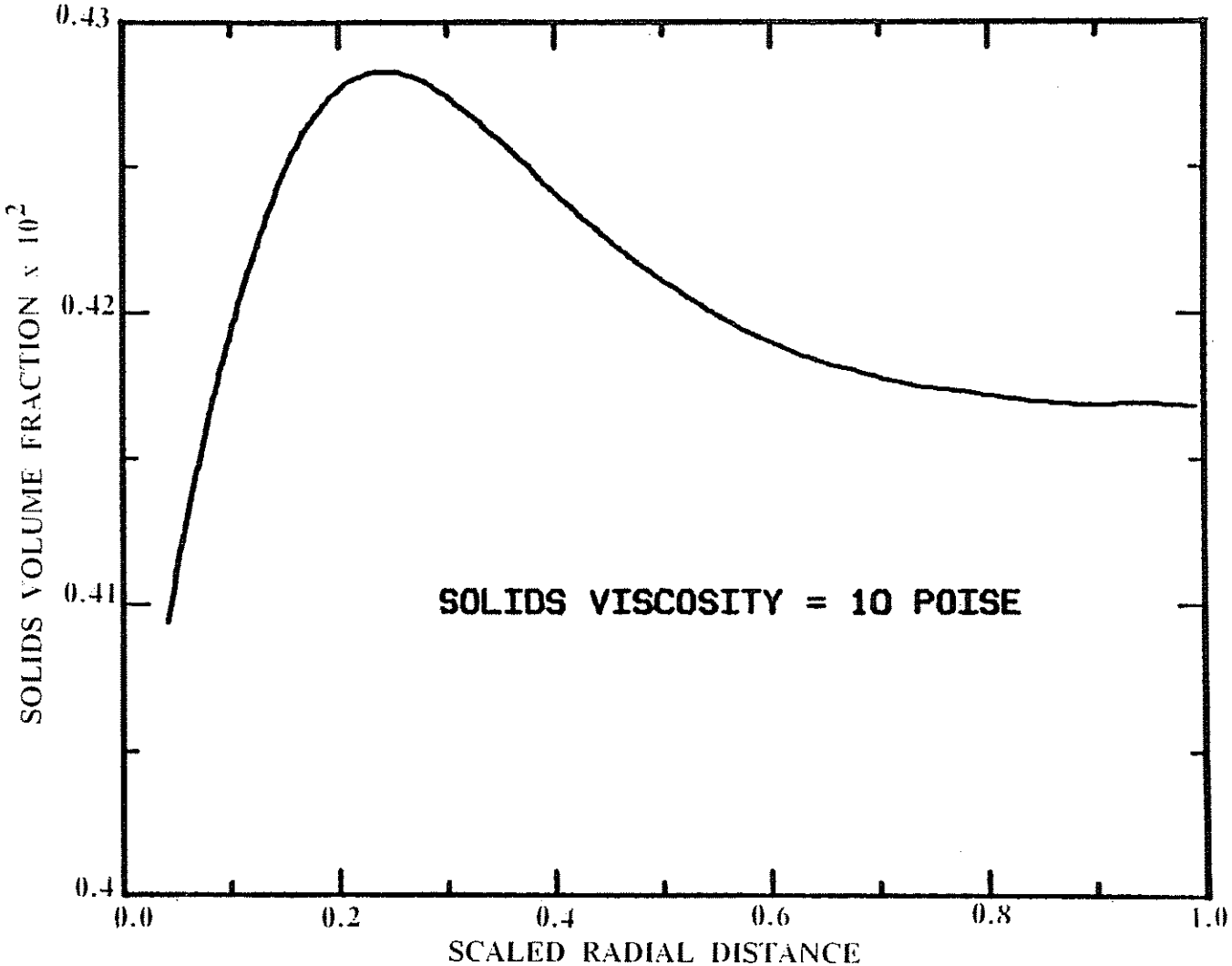


Figure 5.8. Solids Volume Fraction Profiles at a Height of 4m Above the Entrance for μ_s = 10 Poise

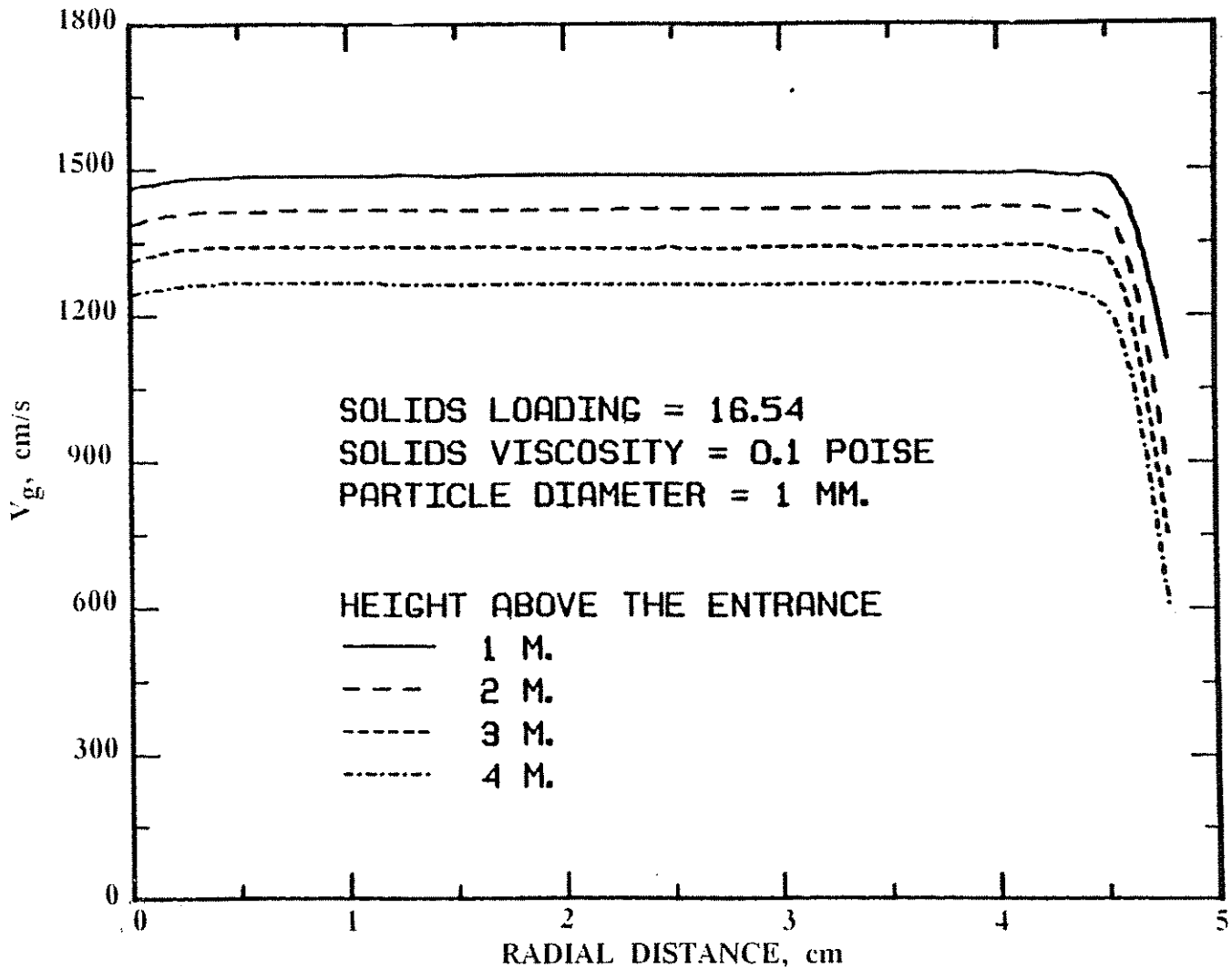


Figure 5.9. Gas Velocity Profiles for a Higher Solids Loading

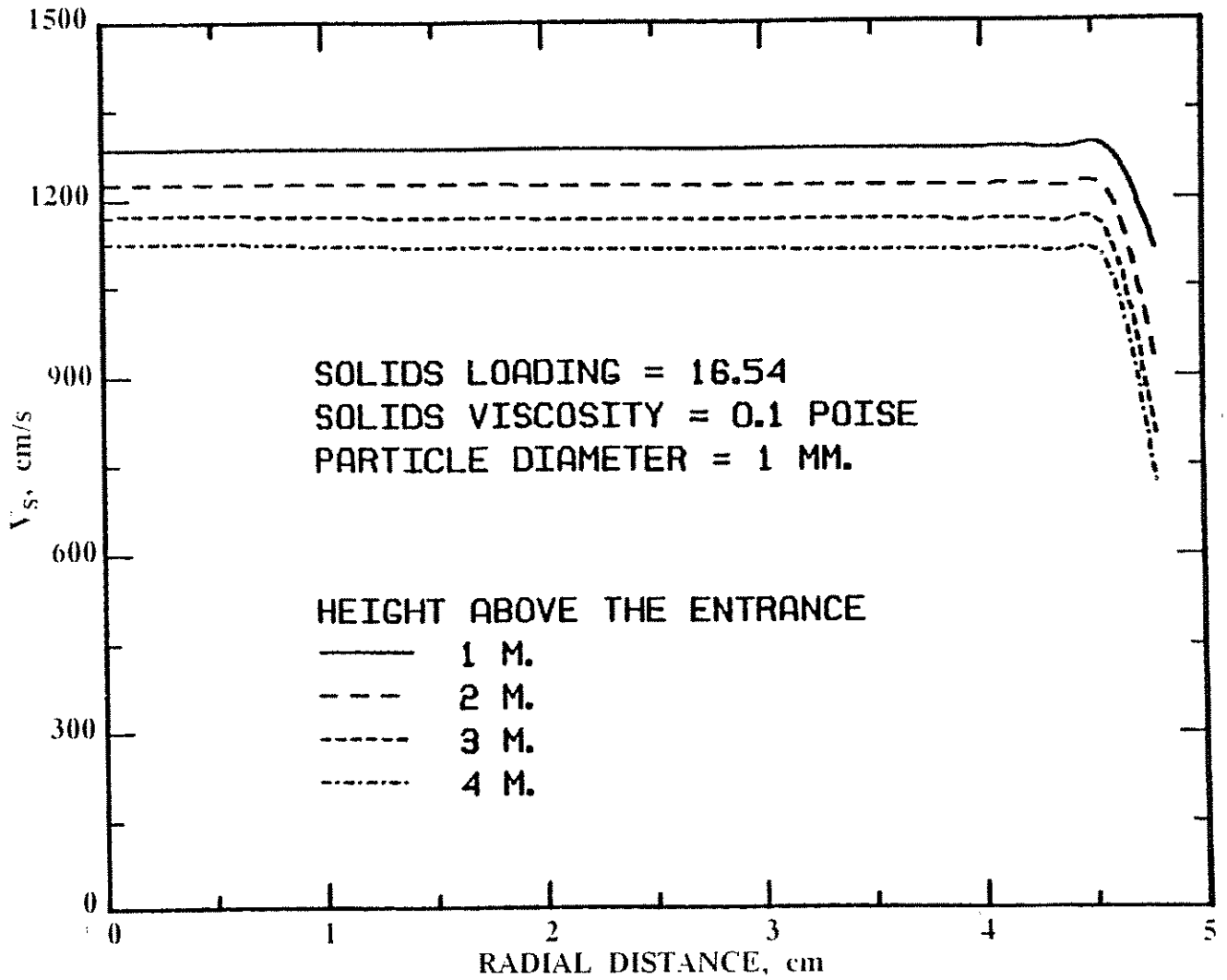


Figure 5.10. Solids Velocity Profiles for a Higher Solids Loading

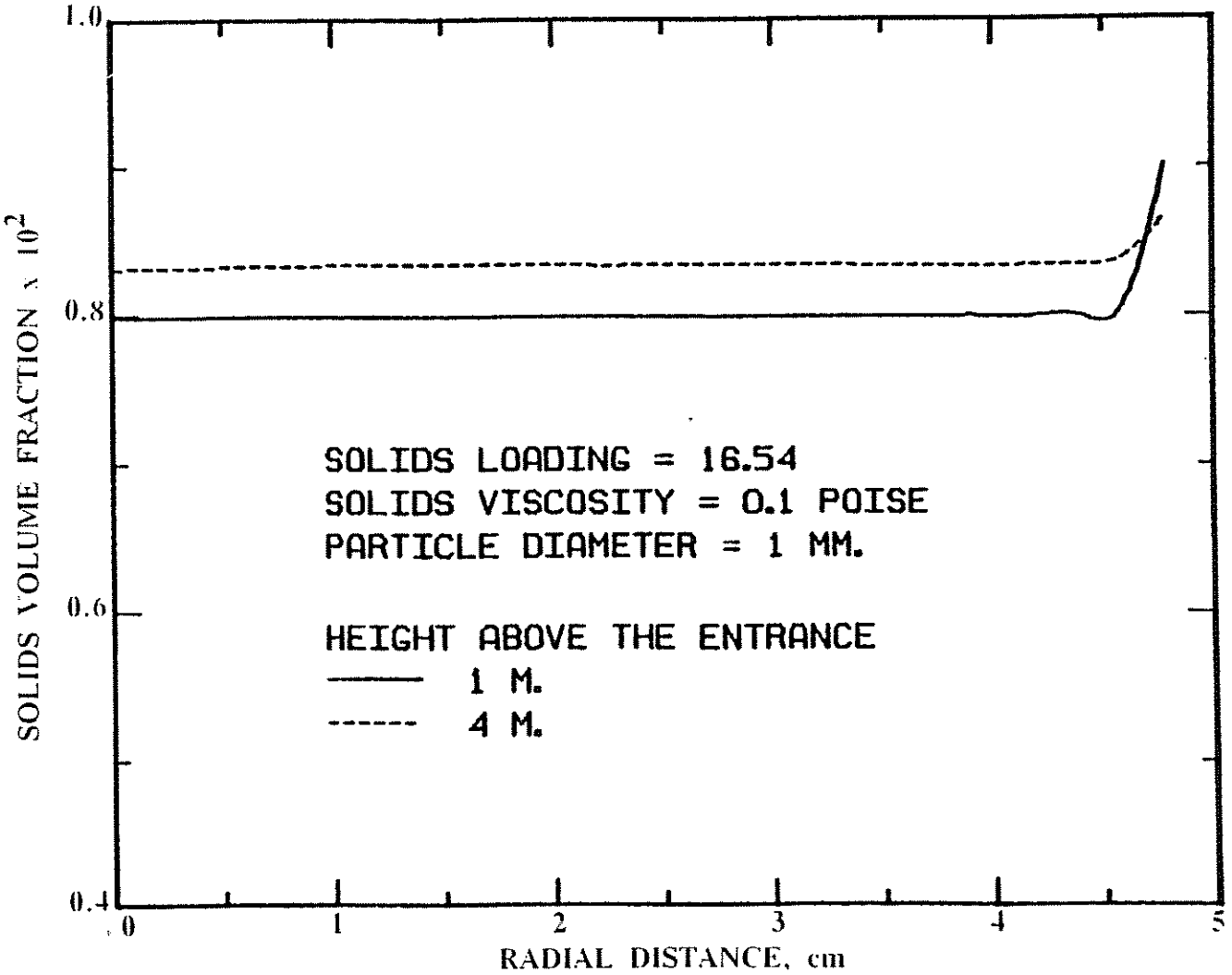


Figure 5.11. Solids Volume Fraction Profiles for a Higher Solids Loading

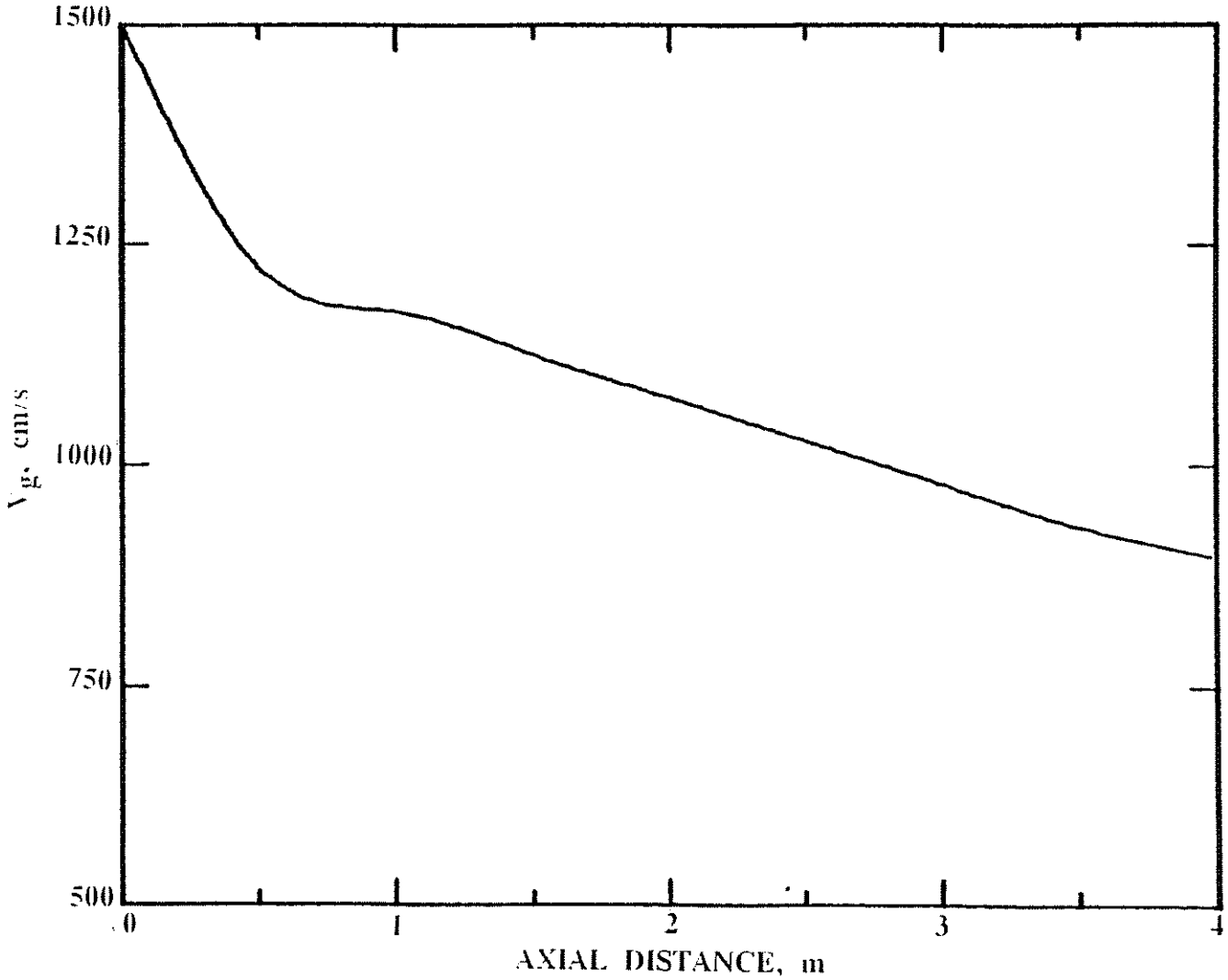


Figure 5.12. Development of the Axial Gas Velocity at the Center

The predicted friction factors are compared to those given by a correlation due to Leung and Wiles (1976), as follows. The friction factor for solids flow is defined as

$$f_s = \frac{\mu_s \, (-\, \partial V_s/\partial r) \text{ wall}}{1/2 \, \rho_s V_s^{\,2}}$$

The correlation recommended by Leung and Wiles (1976) is

$$f_s = 0.05 \, \overline{V}_s^{\,1} \quad (\overline{V}_s \text{ in m/s})$$

The calculations are summarized in Table 5.2

Table 5.2 Comparison of Two Pneumatic Conveyor Simulations

	Simulation: 1	Simulation: 2
μ_s(g/cm s)	1	10
ρ_s(g/cm)	2.9	2.9
$\overline{V}_s$(cm/s)	1200	900
$(-\, \partial Vs/\partial r)_{wall}$ (1/s)	2100	700
Predicted f_s	1×10^{-3}	6×10^{-3}
Correlation f_s	4.1×10^{-3}	5.6×10^{-3}

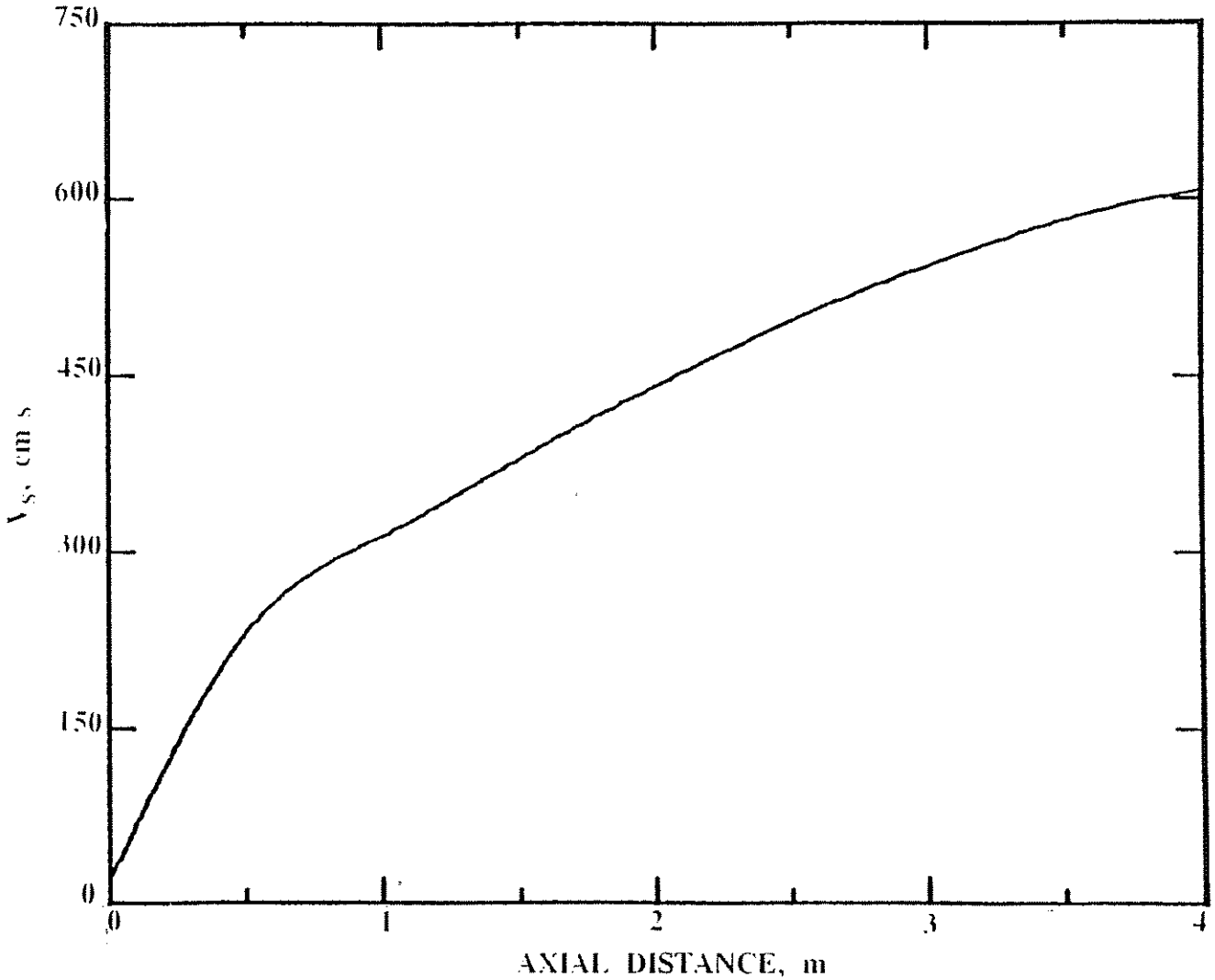


Figure 5.13. Development of the Axial Solids Velocity at the Center

From Table 5.2 a solids viscosity of 10 poise appears to be more appropriate. However, it should be noted that the calculations given above are only approximate.

Segregation in a Pneumatic Conveyor

The equations developed in Chapter II, in one dimension and two particle sizes were used to simulate a segregation experiment due to Nakamura and Capes (1976).

Nakamura and Capes (1976) studied the pneumatic conveying of several binary mixtures and proposed an algebraic equation model to predict the observed segregation of the finer particles. The segregation of the finer material was measured as a ratio of its weight fraction in the test section to its weight fraction at the inlet. Usually this ratio is less than one, since the finer particles move faster than the larger ones. The algebraic equation model of Nakamura and Capes (1976) predicts this phenomena reasonably well. Arastoopour, Lin, and Gidaspow (1980) used three hydrodynamic models to predict the pressure drop and segregation in a similar situation. It was found that the pressure drop could be predicted very well by using the correct value of inlet void fraction. However, the segregation was being over predicted. As correctly observed by them, the discrepancy was due to the absence of binary collisional friction forces in their ordinary differential equation model. Arastoopour, Lin, and Weil (1982) corrected the equations by introducing a particle-particle interaction force of the form used by Nakamura and Capes (1976). The segregation could then be predicted well.

The simulation conditions are shown in Figure 5.14. The initial and boundary conditions used for the simulation are as given below.

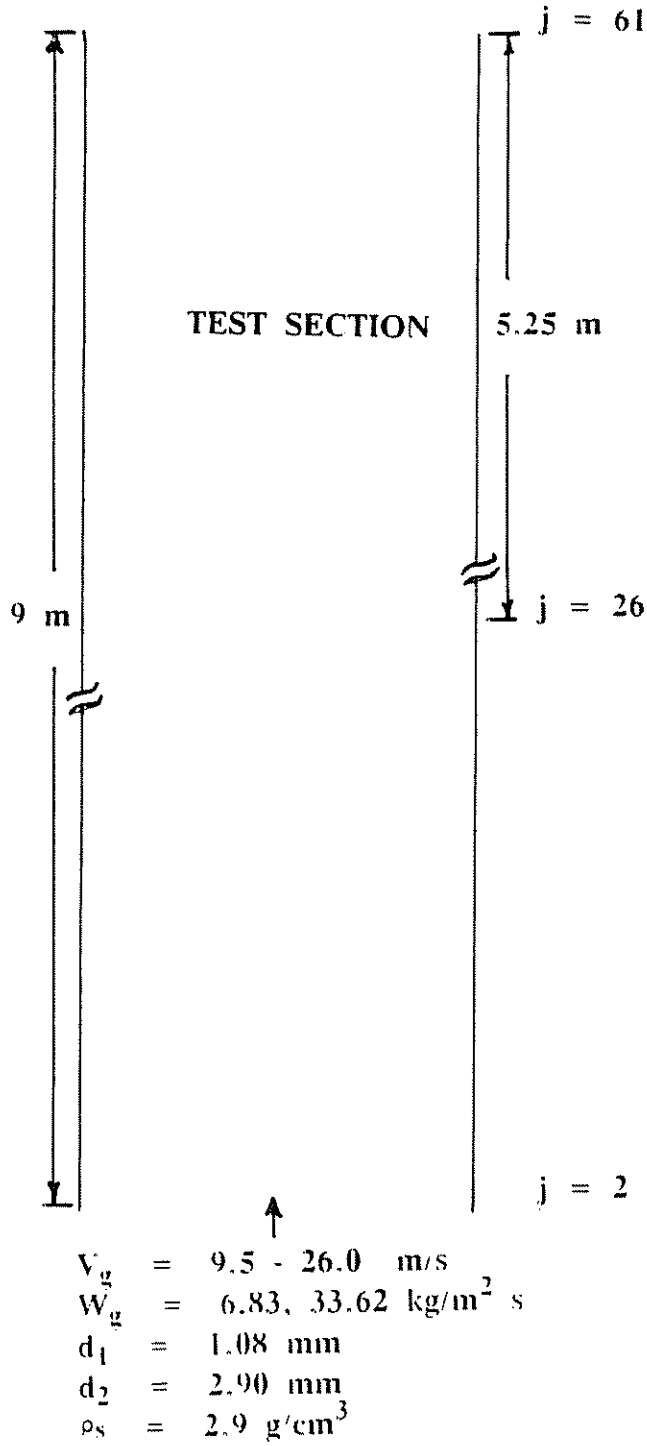


Figure 5.14. Pneumatic Conveying of a Binary Mixture

B.C.1. At $y = 0$, $\rho_g V_g = C$, that is we prescribe a constant mass flux. Also v_2, ε_2, v_3, and ε_3 are prescribed. B.C.2. At $y = L$, we have P = atmospheric. I.C. Uniform conditions similar to B.C.1 are prescribed through out the column. It should be noted that relatively large solids velocities were specified at the inlet to facilitate rapid acceleration of the particles to their final velocities. This was necessary due to the coarse grids used for simulation. In Figure 5.15 the solids velocities at different gas flow rates are shown along the length of the column. It can be seen that the acceleration takes place quite rapidly along the column and that in the test section the velocity is fully developed. The average segregation for a time interval is calculated as follows.

$$\text{Average Segregation} = \frac{\sum_{j=26}^{61} \varepsilon_{2j} / \sum_{j=26}^{61} (1 - \varepsilon_{1j})}{\varepsilon_{2j}/(1 - \varepsilon_{1j})}$$

Figure 5.16 shows the predicted average segregation as a function of gas flow rate. The agreement at high flow rates is good. At low flow rates the gas velocity is smaller than the terminal velocity of the larger particles. However, they are conveyed due to the interaction with the fine particles. The predictions of the model in this region are poor, possibly due to the lack of convergence. In Figure 5.17 the solids velocities have been plotted with the experimentally measured average solids velocity. The agreement with the experiment is reasonably good.

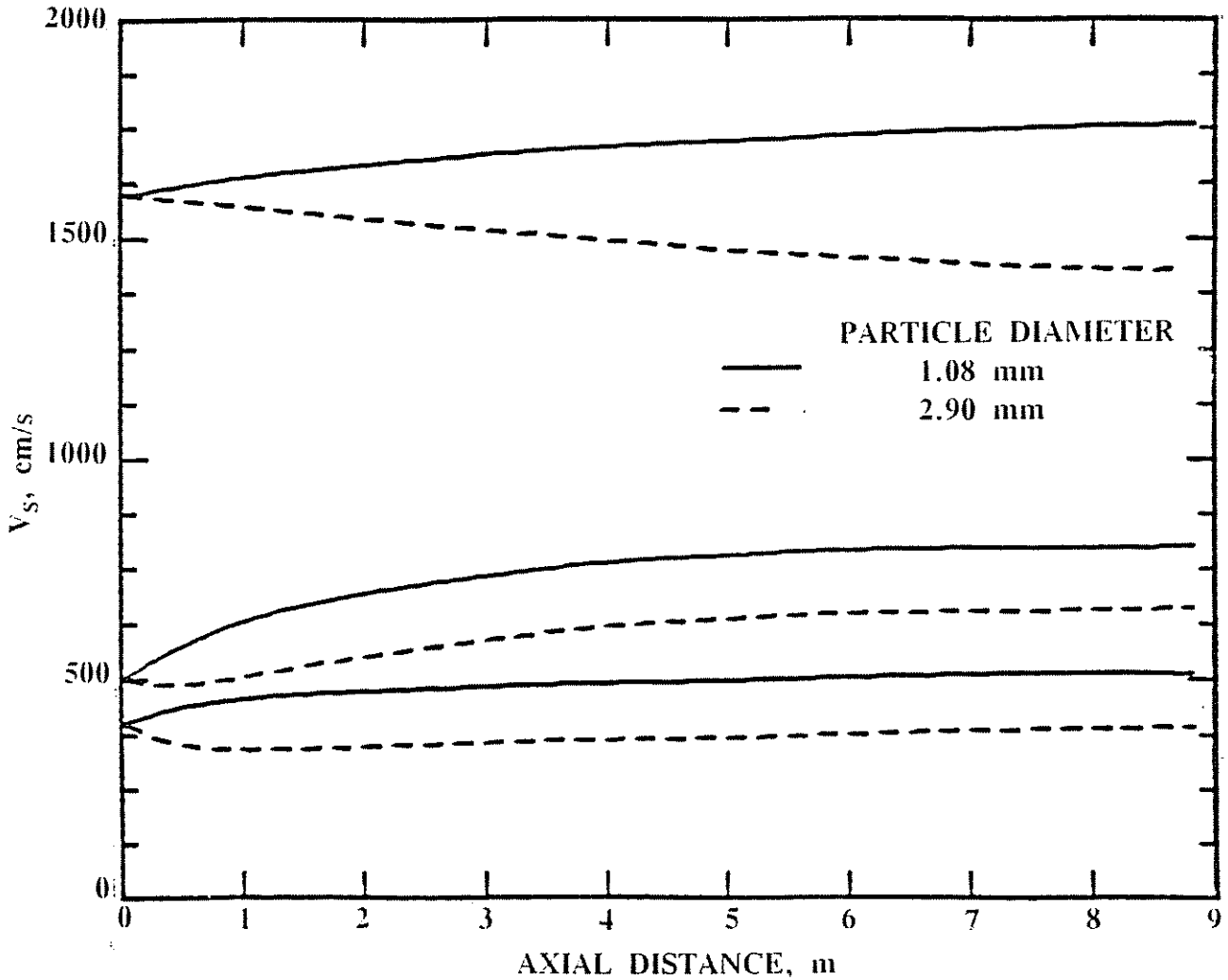


Figure 5.15. Solids Velocity Along the Length of a Pneumatic Conveyor

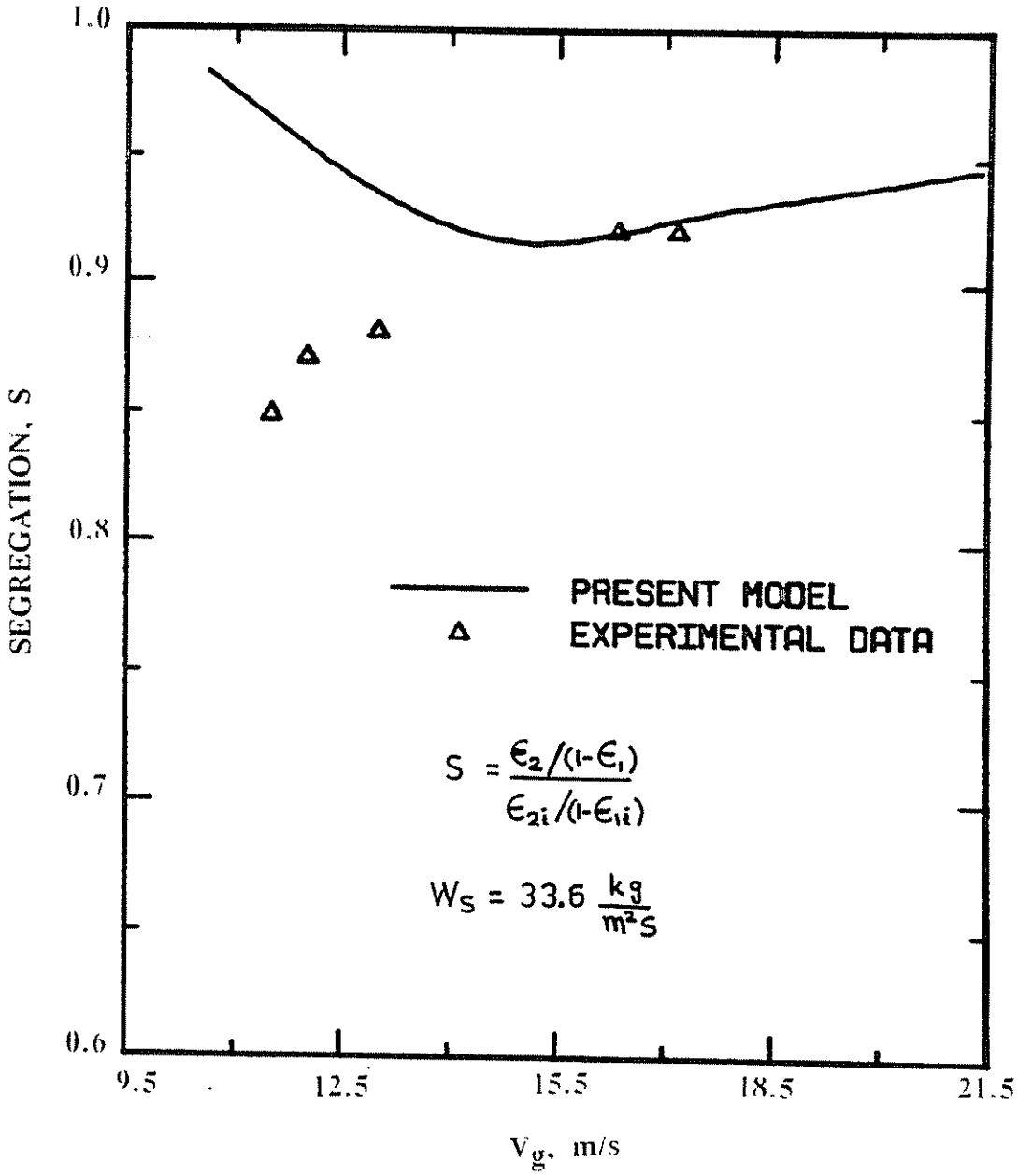


Figure 5.16. Segregation in a Pneumatic Conveyor as a Function of the Gas Velocity

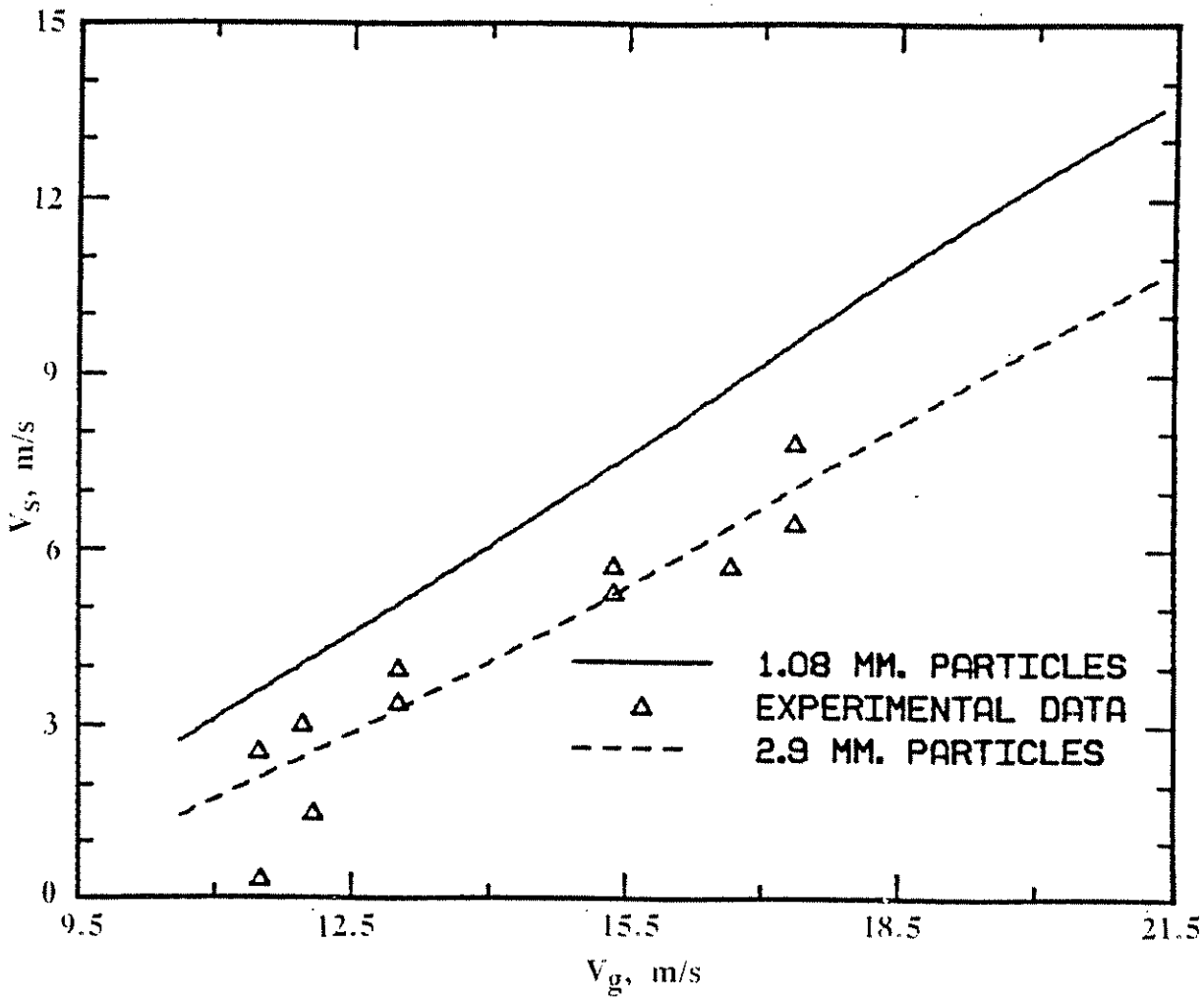


Figure 5.17. Comparison of the Experimental and the Computed Solids Velocities in a Pneumatic Conveyor

CHAPTER VI

CRITICAL MULTIPHASE FLOW

In the previous chapters we looked at numerical simulations based on the multiphase flow equation that predicts an important flow behavior known as critical flow. The understanding and prediction of critical multiphase flow is important in many new energy technologies. For example, critical flow determines the rate of coolant loss in hypothetical accidents in pressurized water and breeder reactors. Critical flow guides the design of pressure protection systems that involve safety-relief valves. Critical flow determines the maximum throughput and choking of coal-fluid mixtures in coal gasifiers and coal liquefaction reactors. Since multiphase critical flow rates are often an order of magnitude smaller than corresponding critical gas flow rates, unexpected and sometimes disastrous results may occur if a design does not properly take into account this substantially different flow rate.

In this chapter we confine our attention to one-dimensional flows and a single particle size. We will also study momentum equations different from those presented in Chapter II.

With the possible exception of some in the Los Alamos group, there appears now to be a concensus that critical flow can be modeled in terms of the vanishing of the determinant (Boure, 1978), explained below. For details, see Rasouli, Gidaspow and Shin (1981).

The conservation of mass, and momentum equations for each phase can be written as:

$$A \frac{\partial U}{\partial t} + B \frac{\partial U}{\partial x} = C \qquad (6.1)$$

where U is a vector of dependent variables such as void fraction, pressure, and gas and liquid velocities, t is time and x the distance in one dimensional flow. A, B and C are functions of U and differ from model to model. The characteristic determinant for the system of equations (6.1) is

$$\left| A\lambda_0 + B\lambda_1 \right| = 0 \tag{6.2}$$

When we set one of the characteristic slopes in Equation (6.2) to zero, some information, such as pressure, cannot propagate upstream. Then, physically, the system governed by Equation (6.1) does not know that, say, the pressure in the reservoir was reduced below a critical pressure. Setting λ_0 to zero in the characteristic polynomial gives

$$\left| B \right| = 0 \tag{6.3}$$

The same condition is obtained from the steady state version of the conservation equations

$$B \frac{\partial U}{\partial x} = C, \tag{6.4}$$

as explained in detail by Boure (1978).

Steady Gas-Solids Flow

We will consider here the three hydrodynamic models discussed by Arastoopour and Gidaspow (1979) for pneumatic transport. However to each equation set we add a solids stress term as discussed in Chapter II.

<u>Annular Flow Model (Equations of Chapter II)</u>. The steady state equations for this case are

$$\begin{bmatrix} \rho_g V_g & \frac{\varepsilon V_g}{C_g^2} & \varepsilon\rho_g & 0 \\ -\rho_s V_s & 0 & 0 & (1-\varepsilon)\rho_s \\ -G & 1 & \varepsilon\rho_g V_g & (1-\varepsilon)\rho_s V_s \\ \frac{-G}{1-\varepsilon} & 1 & 0 & \rho_s V_s \end{bmatrix} \begin{bmatrix} d\varepsilon/dx \\ dp/dx \\ dV_g/dx \\ dV_s/dx \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -(\rho_s(1-\varepsilon) + \varepsilon\rho_g)g \\ \frac{\beta(V_g - V_s)}{1-\varepsilon} - \rho_s g \end{bmatrix} \tag{.65}$$

where G is the "elastic" modulus due to the particle to particle force. The critical condition as given by equation (6.3) in this case is

$$V_s^2 = \frac{G}{\rho_s} + \frac{(1-\varepsilon)\rho_g V_g^2}{\varepsilon\,\rho_s\,(\frac{V_g^2}{C_g^2} - 1)} \tag{6.6}$$

The two limiting cases are

i) $V_g = 0$ or flow in a vacuum, $\rho_g = 0$ giving

$$V_s = \sqrt{\frac{G}{\rho_s}} \tag{6.7}$$

ii) When $\varepsilon - 1$ equation (6.6) correctly reduces to the condition

$$V_g = C_g \tag{6.8}$$

that is for a pure gas we get chocking at the sonic flow rate if equation (6.6) is satisfied at the exit of a pipe. But for choking to

take place within the pipe (or within the domain of equation (6.5)) a compatibility condition as given below should be satisfied (Boure, 1978).

$$\begin{vmatrix} 0 & \frac{\varepsilon V_g}{C_g^2} & \varepsilon\rho_g & 0 \\ 0 & 0 & 0 & (1-\varepsilon)\rho_s \\ -(\rho_s(1-\varepsilon) + (\varepsilon\rho_g)g & 1 & \varepsilon\rho_g V_g & (1-\varepsilon)\rho_s V_s \\ \frac{\beta(V_g-V_s)}{1-\varepsilon} - \rho_s g & 1 & 0 & \rho_s V_s \end{vmatrix} = 0 \qquad (6.9)$$

Upon simplifying equation (6.9) we get

$$\beta(V_g-V_s)\left(1 - \varepsilon \frac{V_g^2}{C_g^2}\right) = (1-\varepsilon)\ \varepsilon\ \rho_s g\left(1 - \frac{V_g^2}{C_g^2}\right) \qquad (6.10)$$

For $V_g << C_g$ equation (6.10) is similar to the minimum fluidization condition. In a situation such as horizontal transport where $g = 0$ the compatibility condition reduces to

$$V_g = V_s \qquad (6.11)$$

implying a homogeneous flow.

<u>Relative Velocity Model</u>. The relative velocity model of Gidaspow (1978) is described by

$$\begin{bmatrix} \rho_g V_g & \frac{\varepsilon V_g}{C_g^2} & \varepsilon\rho_g & 0 \\ -\rho_s V_s & 0 & 0 & \rho_s(1-\varepsilon) \\ -G & 1 & \varepsilon\rho_g V_g & (1-\varepsilon)\rho_s V_s \\ \frac{-G}{[\rho_s(1-\varepsilon)(V_g - V_s)]} & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} d\varepsilon/dx \\ dp/dx \\ dV_g/dx \\ dV_s/dx \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -[\rho_s(1-\varepsilon) + \rho_g\varepsilon]\, g \\ \frac{\beta}{\varepsilon p_s(1-\varepsilon)} \quad \frac{1}{(V_g - V_s)} \end{bmatrix} \tag{6.12}$$

The critical condition in this case is given by

$$\varepsilon\rho_g\left(\frac{G}{V_g - V_s}\right) + \rho_s V_s)\ \left(1 - \frac{\varepsilon V_g^2}{C_g^2}\right) + \frac{\rho_s\varepsilon(1-\varepsilon)V_g}{C_g^2}\ (G - \rho_s V_s^2)$$

$$+ \rho_s(1-\varepsilon)\rho_g V_g = 0 \tag{6.13}$$

The two limiting conditions give similar results as in the previous case.

i) $V_g = 0$ gives $V_s = \sqrt{\frac{G}{\rho_s}}$ (6.14)

ii) $\varepsilon = 1$ gives $V_g = C_g$ (6.15)

The compatibility condition in this case is given by

$$-\beta(V_g - V_s)\ \left(1 - \frac{\varepsilon V_g^2}{C_g^2}\right) + g\ \rho_s\varepsilon(1-\varepsilon)\left(1 - \varepsilon\frac{V_g^2}{C_g^2}\right)$$

$$+ \frac{(\rho_s(1-\varepsilon) + \rho_g\varepsilon)\ g\ V_g(V_g - V_s)\ \varepsilon(1-\varepsilon)\rho_s}{C_g^2\ \rho_g} = 0 \tag{6.16}$$

If $V_g << C_g$ equation (6.16) simplifies to the minimum fluidization condition

$$\beta(V_g - V_s) = g\ \rho_s\ \varepsilon(1-\varepsilon) \tag{6.17}$$

and when g = 0, it gives

$$V_g = V_s$$

implying a homogeneous flow.

<u>Pressure Drop in Fluid Phase Model</u>. The equation describing this model is

$$\begin{bmatrix} \rho_g V_g & \frac{\varepsilon V_g}{C_g^2} & \varepsilon\rho_g & 0 & d\varepsilon/dx \\ -\rho_s V_s & 0 & 0 & (1-\varepsilon)\rho_s & dp/dx \\ -G & 1 & \varepsilon\rho_g V_g & (1-\varepsilon)\rho_s V_s & dV_g/dx \\ \frac{-G}{1-\varepsilon} & 0 & 0 & \rho_s V_s & dV_s/dx \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -(\rho_s(1-\varepsilon) + \varepsilon\rho_g)g \\ \frac{\beta(V_g - V_s)}{\varepsilon(1-\varepsilon)} - \rho_s g \end{bmatrix} \tag{6.19}$$

The critical condition is given by

$$(G - \rho_s V_s^2)\ (1 - \frac{\varepsilon^2 g}{C_g^2}) = 0 \tag{6.20}$$

which reduces to the same limiting conditions as in the case of the previous models. This compatibility condition is given by

$$(1-\varepsilon)\ (\frac{\beta(V_g - V_s)}{\varepsilon(1-\varepsilon)} - \rho_s g)\ (1 - \frac{\varepsilon V_g^2}{C_g^2}) = 0 \tag{6.21}$$

which also reduces to the same limiting conditions as in the case of the previous models.

$\sqrt{G/\rho_s}$ critical condition.

In absolute vacuum (i.e. p=0) all three models reduce themselves to

$$\begin{bmatrix} \rho_s V_s & \varepsilon_s \rho_s & d\varepsilon_s/dx \\ \frac{G}{\varepsilon_s} & \rho_s V_s & dV_s/dx \end{bmatrix} = \begin{bmatrix} 0 \\ -\rho_s g \end{bmatrix} \qquad (6.22)$$

from where it can be easily seen that $\sqrt{G/\rho_s}$ is the speed of propagation of void fraction disturbances in a bed of particles. Thus in a choked granular flow the discharge rate is limited by this propagation velocity.

Discharge From a Hopper

Altiner and Davidson (1980) studied the discharge rate from an aerated hopper for three particle sizes (159, 276 and 390 microns diameter) and found the superficial solids velocity $\left(V_s(1-\varepsilon)\right)$ at $V_g = 0$ to be approximately 17 cm/s. Assuming that the flow is choked at the exit of the hopper, the solids velocity will be governed by $\sqrt{G/\rho_s}$. G is given by

$$G = 1.5\ 10^{-3} \exp\left((0.44 - \varepsilon)500\right) \text{ dynes/cm}^2 \qquad (6.23)$$

and

$$\rho_s = 2.6 \text{ gm/cm}^3$$

Now using a value of $\varepsilon = 0.411$ we get

$$V_s(1-\varepsilon) = 19.9 \text{ cm/s}$$

which compares well with the experimental value. However, further study is necessary to get a correct form of G.

Critical Flow From a Fluidized Bed

The multiphase flow model developed here was used to calculate the discharge rate of solids from a pressurized fluidized bed. The computational region is as shown in Figure 6.1.

Computational difficulties were anticipated due to the rapid change in the flow field near the orifice. Usage of a very fine spatial noding is, however, not a practical solution in this case (Trapp and Ransom, 1982). The preliminary computations could be carried out only up to 8m/s. The results are summarized in Figure 6.2. The histories of gas and solid velocity profiles indicate the possibility of choking occurring at the orifice. Hancox et al. (1980) got similar velocity profiles with a maximum at 2m/s for a blow down problem. They attribute the maxima to a choking phenomenon. In our calculations, however, we could not see the choking condition as given by equation (6.6) being satisfied in the duration of 8m/s. It is possible that the present computer code could not handle the onset of choking well. Computational difficulties may be removed by using the characteristic directions to handle the boundaries.

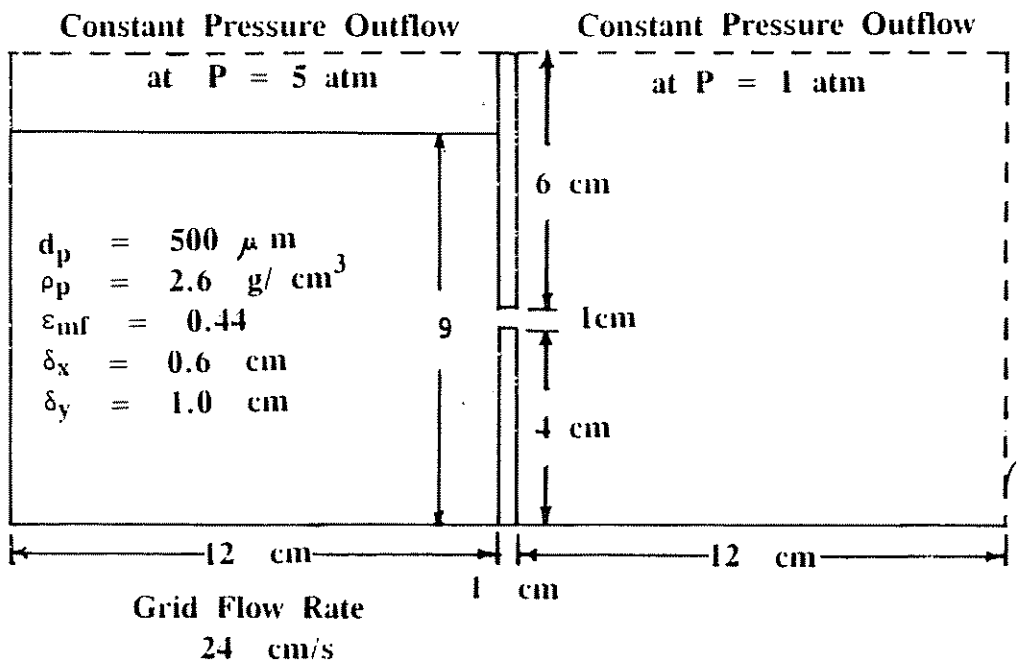


Figure 6.1. Computational Region for the Critical Flow Calculations

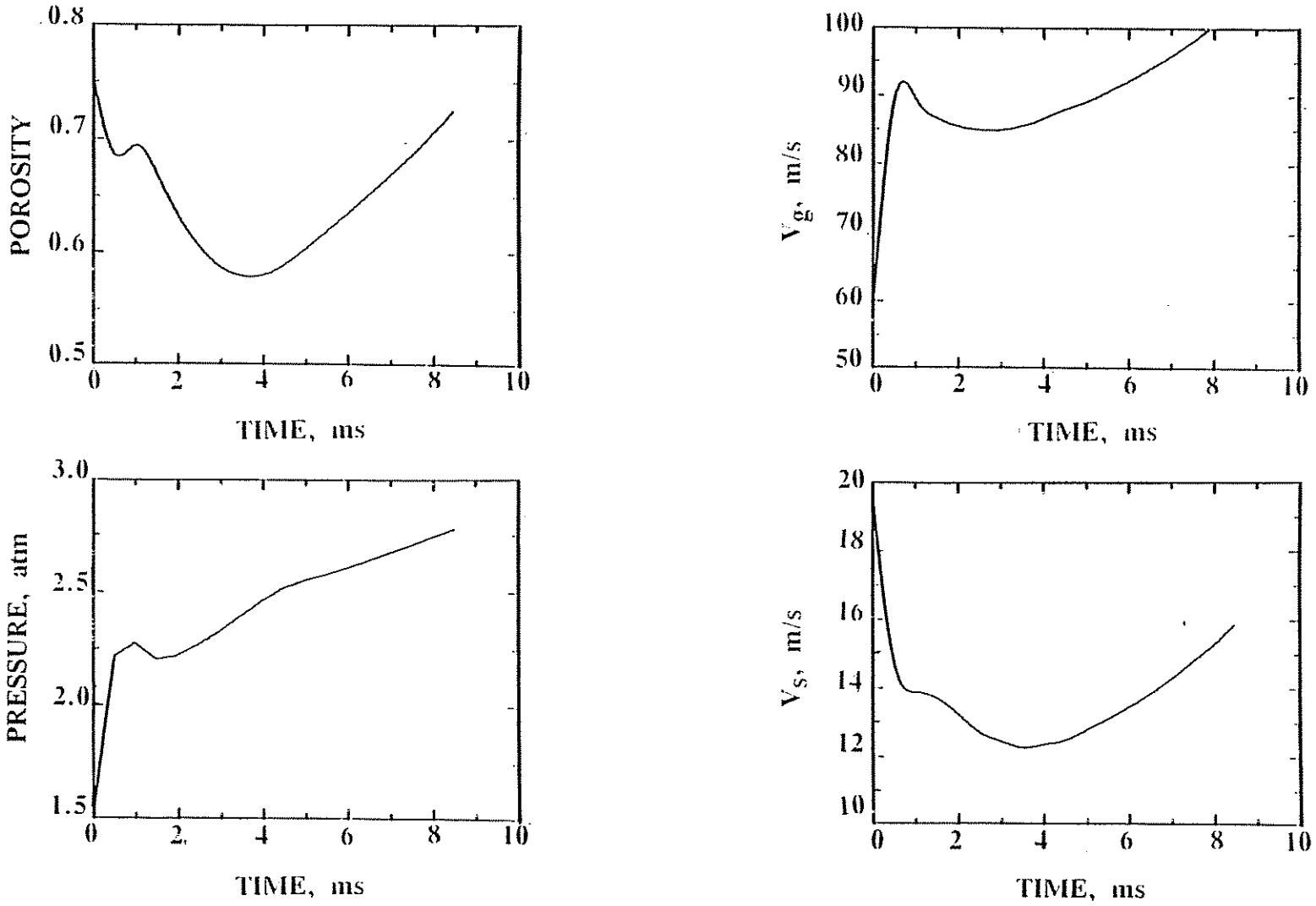


Figure 6.2. Transient Behavior of the Local Variables at the Orifice

CHAPTER VII

SUMMARY AND RECOMMENDATIONS

The central theme of this thesis has been the application of multiphase flow models to dense and dilute phase gas-solids flows. General equations for a mixture of particles were developed. In these equations similar particles were assumed to form a continuum — a particulate phase. A computer model has been developed based on these equations.

To model systems with more than one particulate phase it is necessary to introduce a particle-particle interaction term. An approximate formula was derived to model this term. The model was applied to several gas-solids flow systems with a binary mixture of particles. Some of these computational results were compared to experimental data with reasonable success.

Fluidized beds are excellent heat transfer media due to the large heat transfer rates. A critical test for a fluidized bed hydrodynamic model is its ability to predict the experimentally observed large heat transfer coefficients. Energy equations were added to the hydrodynamic model and it was verified that the model predicts large heat transfer coefficients which compared favorably with experimental data.

A viscosity term was added to model radial velocity profiles in a pneumatic conveyor. The velocity profiles were similar to one of two contradicting sets of published experimental data. The computations predicted nearly flat velocity profiles. One-dimensional simulation to

study segregation in a pneumatic conveyor for a binary mixture was carried out. The results were compared to experimental data.

Three multiphase flow models were studied to determine critical flow conditions. It was found that dense phase transport may be limited by a critical velocity equal to $\sqrt{G/\rho_s}$ where G is an "elastic" modulus of the powder and ρ_s is its density.

The computer program developed here can be used to study a variety of problems. Unfortunately, it is not fully validated and optimized for production mode runs. Hence, the following steps seem to be necessary for further development:

1. Some experimental study is necessary to ascertain the nature of the solids stress term.
2. Comparison of the theoretical predictions with the instantaneous solids volume fraction and the solids velocity data in the mixing layer should be made to validate the constitutive relation used for particle-particle interaction.
3. Simulation of a fast bubble has not been made so far, which is necessary for a full validation of the hydrodynamic model for fluidization.
4. The bubbles predicted at present have a smeared boundary. Incorporation of some method similar to the volume of fluid method (Hirt and Nichols, 1981) to get a sharper bubble boundary should be explored. Another possibility is to shift the location of the solids volume fraction from the cell center to the cell boundaries to properly handle the jump in velocity.

5. In future computations the assumption of symmetry at the center should be avoided since it is not a correct boundary condition.

6. The program presented here is in modular form and lacks optimization. Also it requires large computational time. There is ample scope for program refinement to improve computational efficiency. A better, but more involved, alternative is to vectorize the program for cost effective computing on a CRAY.

7. The program in its present form cannot march quickly towards a steady state. Through an implicit treatment of the convective terms this could be rectified.

8. The program permits inflow from bottom and left boundaries only and out flow from the top and right boundaries. It is suspected that the particular computational sweep used in the program may be introducing a bias in the flow directions. This problem needs some attention.

APPENDIX A

A MANUAL FOR THE MULTIFIX PROGRAM

The MULTIFIX program solves a set of partial differential equations to describe the flow of a gas–solids mixture. The program allows the description of the flow of solid particles of different sizes. Each size of particles is treated as a continuum— a particulate phase, with a characteristic diameter.

The transient solution is carried out in a two–dimensional region. Rectangular, cylindrical or one–dimensional spherical coordinates may be used to set up a problem.

The program can be applied to a variety of gas-solids flows of practical interest such as fluidization,pneumatic conveying, etc. It can also be used to liquid-solids flows and with some modifications to gas-liquid-solids flows.

The numerical method is based on the one due to Harlow and Amsden (1975) which was subsequently used in the K-FIX program (Rivard and Torrey, 1977). Earlier a single particle size version was used to study fluidization (Gidaspow and Ettehadieh, 1983).

THE DIFFERENTIAL EQUATIONS

The differential equations are written in a form suggested by Soo(1967). The continuity and momentum equations are written for the gas phase and each of the particulate phases. Note that the subscript "1" is used to indicate gas phase.

Continuity Equations

$$\frac{\partial}{\partial t}(\rho_k \epsilon_k) + \nabla.(\epsilon_k \rho_k \mathbf{v}_k) = 0 \tag{8.1}$$

Momentum Equations

$$\begin{aligned}\frac{\partial}{\partial t}(\rho_k \epsilon_k \mathbf{v}_k) + \nabla.(\epsilon_k \rho_k \mathbf{v}_k \mathbf{v}_k) &= -\epsilon_k \nabla P + \rho_k \epsilon_k \mathbf{g} + G_k \nabla \epsilon_1 \\ &\quad + \sum_{l=1}^{N} K_{kl}(\mathbf{v}_l - \mathbf{v}_k)\end{aligned} \tag{8.2}$$

where,

$$\sum_{k=1}^{N} \epsilon_k = 1 \tag{8.3}$$

Equation of State

$$\begin{aligned} \rho_1 &= \frac{P}{RT} \\ \rho_k &= \rho_{sk} \qquad \text{for} k \geq 2 \end{aligned} \tag{8.4}$$

These equations are supplemented by the constitutive relations for the interphase momentum transfer and the solids stress.

Gas–Particle Friction

$$\begin{aligned} K_{1k} = K_{k1} &= \frac{150(1-\epsilon_1)\epsilon_k \mu_1}{\epsilon_1 (d_k \phi_k)^2} + \frac{1.75 \rho_1 |\mathbf{v}_1 - \mathbf{v}_k| \epsilon_k}{(d_k \phi_k)} \qquad 0.2 \leq \epsilon_1 < 0.8 \\ &= \frac{3}{4} C_{Dk} \frac{\epsilon_1 |\mathbf{v}_1 - \mathbf{v}_k| \rho_1 \epsilon_k}{(d_k \phi_k)} f(\epsilon_1) \qquad 0.8 \leq \epsilon_1 \leq 1.0 \end{aligned} \tag{8.5}$$

where,

$$f(\epsilon_1) = \epsilon_1^{-2.65}, \tag{8.6}$$

$$\begin{aligned} C_{Dk} &= \frac{24}{Re_k}\left(1 + 0.15 Re_k^{0.687}\right) \qquad Re_k < 1000 \\ &= 0.44, \qquad Re_k \geq 1000 \end{aligned} \tag{8.7}$$

and

$$Re_k = \frac{d_k |\mathbf{v}_1 - \mathbf{v}_k| \rho_1 \epsilon_1}{\mu_1} \tag{8.8}$$

Solids Stress

$$\begin{aligned} G_k &= 0 \qquad \text{for } k = 1 \\ &= g_1 \epsilon_k \, exp\left[g_2\left(\epsilon^* - \epsilon_1\right)\right] \qquad \text{for } k = 2, \ldots, N \end{aligned} \tag{8.9}$$

Particle-Particle Friction

$$K_{\substack{kl \\ k,l\neq 1}} = F(1+e)\frac{\rho_k\rho_l\epsilon_k\epsilon_l(d_k+d_l)^2}{\rho_k d_k^3+\rho_l d_l^3}|\mathbf{v}_k-\mathbf{v}_l| \tag{8.10}$$

where

$$F = \frac{3\epsilon_{kl}^{\frac{1}{3}}+(\epsilon_k+\epsilon_l)^{\frac{1}{3}}}{4\left(\epsilon_{kl}^{\frac{1}{3}}-(\epsilon_k+\epsilon_l)^{\frac{1}{3}}\right)} \tag{8.11}$$

$$\begin{aligned}\epsilon_{kl} &= [(\Phi_k-\Phi_l)+(1-a)(1-\Phi_k)\Phi_l][\Phi_k+(1-\Phi_k)\Phi_l]\frac{X_k}{\Phi_k}+\Phi_l \\ &\qquad \text{for} X_k \leq \frac{\Phi_k}{\Phi_k+(1-\Phi_k)\Phi_l} \\ &= (1-a)[\Phi_k+(1-\Phi_k)\Phi_l](1-X_k)+\Phi_k \\ &\qquad \text{for} X_k \geq \frac{\Phi_k}{\Phi_k+(1-\Phi_k)\Phi_l}\end{aligned} \tag{8.12}$$

$$a = \sqrt{\frac{d_l}{d_k}} \qquad \text{when} d_k \geq d_l \tag{8.13}$$

$$X_k = \frac{\epsilon_k}{\epsilon_k+\epsilon_l} \tag{8.14}$$

Φ_k, Φ_l – volume fractions at maximum packing

These constitutive relations may be easily changed by some program modifications.

The problem definition is complete with the specification of appropriate boundary and initial conditions. The program has the capability of treating a variety of boundary and initial conditions.

THE FINITE DIFFERENCE EQUATIONS

The computations are carried out using a mesh of finite difference cells fixed in a two-dimensional space (Eulerian Mesh). A typical computational cell is shown in Figure 8.1. The scalar variables are located at the cell center and the vector variables at the cell boundaries.

The continuity equation is differenced fully implicitly as follows.

$$
\begin{aligned}
{}^{n+1}(\epsilon_k\rho_k)_i^j &= {}^{n}(\epsilon_k\rho_k)_i^j - \frac{\delta t}{r_i\delta r}\,{}^{n+1}\langle(\epsilon_k\rho_k r)u_k\rangle_i^j \\
&\quad - \frac{\delta t}{\delta z}\,{}^{n+1}\langle(\epsilon_k\rho_k)v_k\rangle_i^j
\end{aligned}
\tag{8.15}
$$

The angular brackets represent a donor cell differencing as shown below.

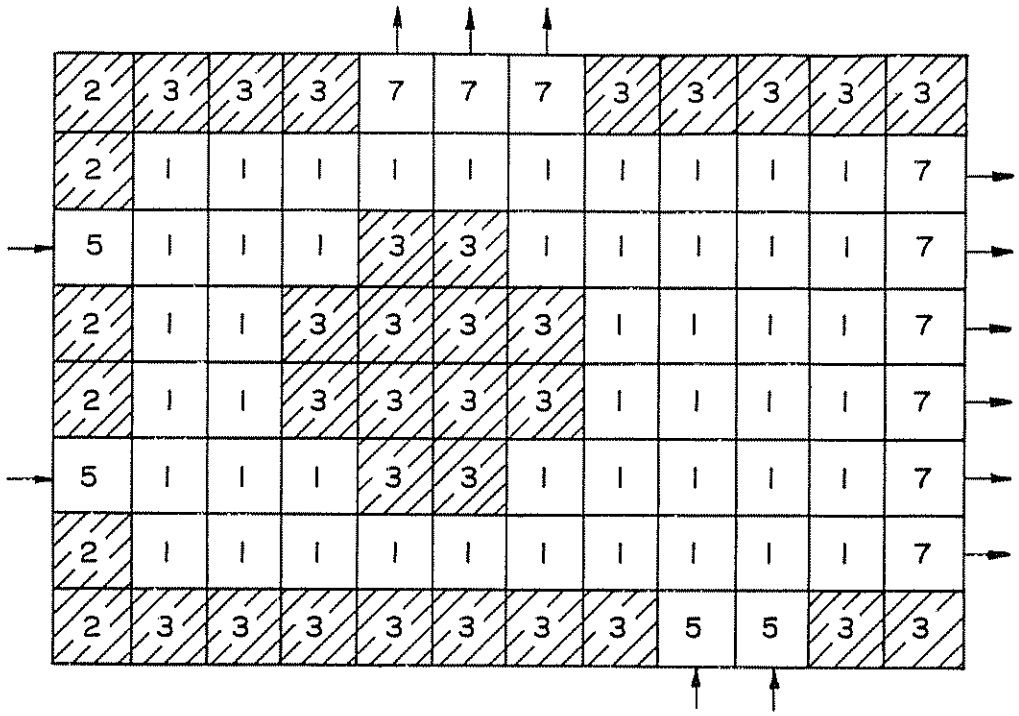
$$
\begin{aligned}
\langle(\epsilon_k\rho_k r)u_k\rangle_i^j &= (u_k)_{i+\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k r)_i\,, & \text{if } (u_k)_{i+\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k r)_{i+1}\,, & \text{if } (u_k)_{i+\frac{1}{2}} < 0.\end{cases}\\
&\quad - (u_k)_{i-\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k r)_{i-1}\,, & \text{if } (u_k)_{i-\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k r)_i\,, & \text{if } (u_k)_{i+\frac{1}{2}} < 0.\end{cases}
\end{aligned}
\tag{8.16}
$$

and

$$
\begin{aligned}
\langle(\epsilon_k\rho_k)v_k\rangle_i^j &= (v_k)^{j+\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k)^j\,, & \text{if } (v_k)^{j+\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k)^{j+1}\,, & \text{if } (v_k)^{j+\frac{1}{2}} < 0.\end{cases}\\
&\quad - (v_k)^{j-\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k)_{i-1}\,, & \text{if } (v_k)^{j-\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k)^j\,, & \text{if } (v_k)^{j+\frac{1}{2}} < 0.\end{cases}
\end{aligned}
\tag{8.17}
$$

The donor cell differencing helps to prevent a cell from getting drained completely giving negative volume fractions and also aids computational stability.

The momentum equations are differenced over a staggered mesh of computational cells (Figure 8.2). The difference equations are



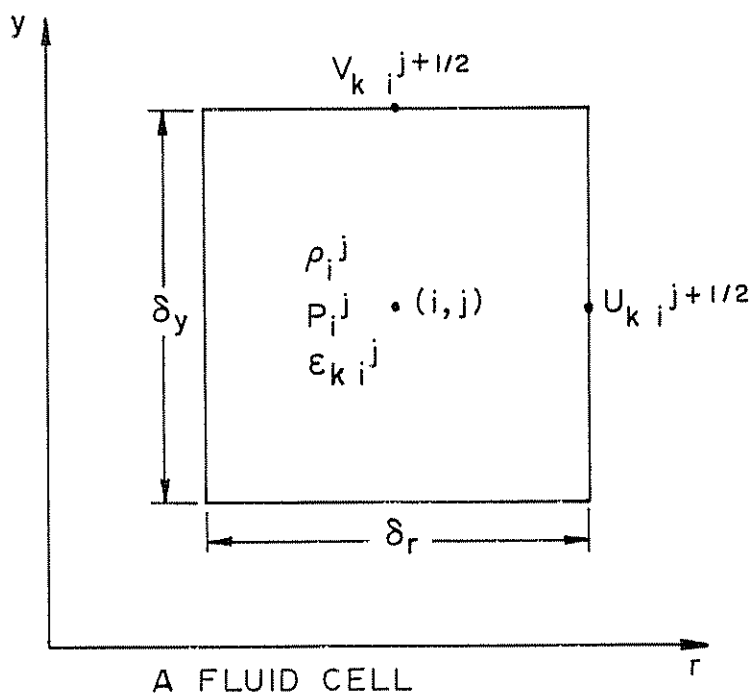
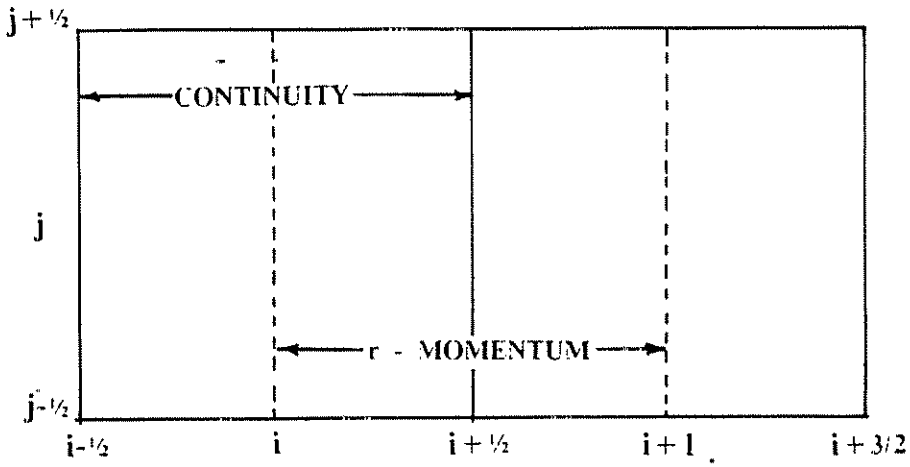


Figure 8.1. The Computational Mesh



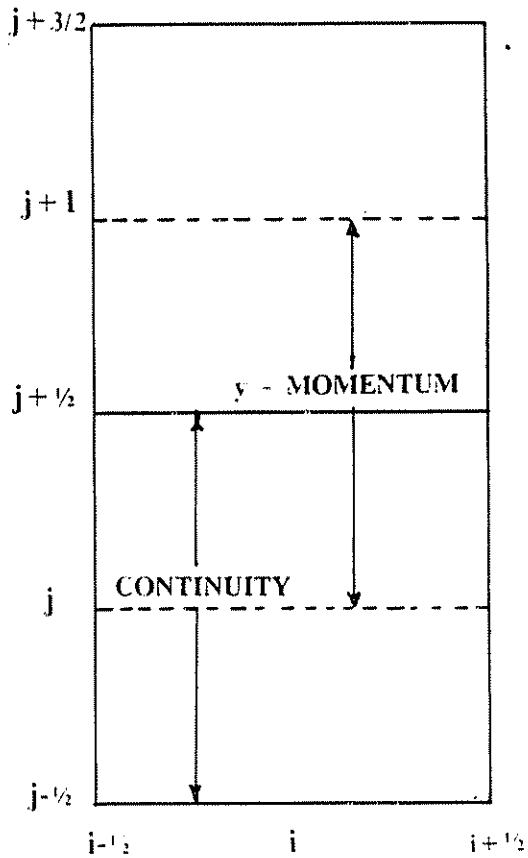


Figure 8.2. The Staggered Computational Mesh for Momentum Equations

$$
\begin{aligned}
{}^{n+1}(\epsilon_k\rho_k u_k)^j_{i+\frac{1}{2}} &= (\overline{\epsilon_k\rho_k u_k})^j_{i+\frac{1}{2}} - \frac{\delta t}{\delta r}\ {}^{n+1}(\epsilon_k)^j_{i+\frac{1}{2}}\ {}^{n+1}(P_{i+1}-P_i)^j \\
&\quad + \delta t\sum_{l=1}^{N}\ {}^{n}K^j_{kl_{i+\frac{1}{2}}}\ {}^{n+1}(u_l-u_k)^j_{i+\frac{1}{2}} \\
&\quad + \frac{\delta t}{\delta r}\ {}^{n+1}(G_k)^j_{i+\frac{1}{2}}\ {}^{n+1}((\epsilon_1)_{i+1}-(\epsilon_1)_i)^j
\end{aligned}
\tag{8.18}
$$

$$
\begin{aligned}
{}^{n+1}(\epsilon_k\rho_k v_k)^{j+\frac{1}{2}}_i &= (\overline{\epsilon_k\rho_k v_k})^{j+\frac{1}{2}}_i - \frac{\delta t}{\delta z}\ {}^{n+1}(\epsilon_k)^{j+\frac{1}{2}}_i\ {}^{n+1}\Big(P^{j+1}-P^j\Big)_i \\
&\quad + \delta t\sum_{l=1}^{N}\ {}^{n}K^{j+\frac{1}{2}}_{kl_i}\ {}^{n+1}(v_l-v_k)^{j+\frac{1}{2}}_i \\
&\quad + \frac{\delta t}{\delta z}\ {}^{n+1}(G_k)^{j+\frac{1}{2}}_i\ {}^{n+1}\Big((\epsilon_1)^{j+1}-(\epsilon_1)^j\Big)_i
\end{aligned}
\tag{8.19}
$$

All the explicit terms have been lumped into the *tilde* quantities as shown below.

$$
\begin{aligned}
(\overline{\epsilon_k\rho_k u_k})^j_{i+\frac{1}{2}} &= {}^{n}(\epsilon_k\rho_k u_k)^j_{i+\frac{1}{2}} - \frac{\delta t}{r_i\delta r}\ {}^{n}\langle(\epsilon_k\rho_k u_k r)u_k\rangle^j_{i+\frac{1}{2}} \\
&\quad - \frac{\delta t}{\delta z}\ {}^{n}\langle(\epsilon_k\rho_k u_k)v_k\rangle^j_{i+\frac{1}{2}}
\end{aligned}
\tag{8.20}
$$

$$
\begin{aligned}
(\overline{\epsilon_k\rho_k v_k})^{j+\frac{1}{2}}_i &= {}^{n}(\epsilon_k\rho_k v_k)^{j+\frac{1}{2}}_i - \frac{\delta t}{r_i\delta r}\ {}^{n}\langle(\epsilon_k\rho_k v_k r)u_k\rangle^{j+\frac{1}{2}}_i \\
&\quad - \frac{\delta t}{\delta z}\ {}^{n}\langle(\epsilon_k\rho_k v_k)v_k\rangle^{j+\frac{1}{2}}_i - \epsilon_k\rho_k g\delta t
\end{aligned}
\tag{8.21}
$$

As explained before the angular brackets represent the donor cell differenced quantities. For example,

$$\langle(\epsilon_k\rho_k u_k)v_k\rangle^j_{i+\frac{1}{2}} = (v_k)^{j+\frac{1}{2}}_{i+\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k u_k)^j_{i+\frac{1}{2}}, & \text{if } (v_k)^{j+\frac{1}{2}}_{i+\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k u_k)^{j+1}_{i+\frac{1}{2}}, & \text{if } (v_k)^{j+\frac{1}{2}}_{i+\frac{1}{2}} < 0.\end{cases}$$
$$-(v_k)^{j-\frac{1}{2}}_{i+\frac{1}{2}}\begin{cases}(\epsilon_k\rho_k u_k)^{j-1}_{i+\frac{1}{2}}, & \text{if } (v_k)^{j-\frac{1}{2}}_{i+\frac{1}{2}} > 0;\\ (\epsilon_k\rho_k u_k)^{j}_{i+\frac{1}{2}}, & \text{if } (v_k)^{j-\frac{1}{2}}_{i+\frac{1}{2}} < 0.\end{cases} \tag{8.22}$$

and

$$\langle(\epsilon_k\rho_k v_k)v_k\rangle^{j+\frac{1}{2}}_i = (v_k)^{j+1}_i\begin{cases}(\epsilon_k\rho_k v_k)^{j+\frac{1}{2}}_i, & \text{if } (v_k)^{j+1} > 0;\\ (\epsilon_k\rho_k v_k)^{j+\frac{3}{2}}_i, & \text{if } (v_k)^{j+1}_i < 0.\end{cases}$$
$$-(v_k)^j_i\begin{cases}(\epsilon_k\rho_k v_k)^{j-\frac{1}{2}}_i, & \text{if } (v_k)^j_i > 0;\\ (\epsilon_k\rho_k v_k)^{j+\frac{1}{2}}_i, & \text{if } (v_k)^j_i < 0.\end{cases} \tag{8.23}$$

In the momentum equations the pressure , the drag and the solids stress terms are implicitly differenced. Also note that the drag terms are linearly implicit which is necessary for the particular method of solution adopted here. There are no other restrictions on the form of the drag relations. In calculations involving dense beds it is necessary to have the solids stress term differenced implicitly. If dense regions are absent in the computational regime this term may be differenced explicitly to save computational time.

Equations (8.3), (8.4), (8.15), (8.18), (8.19) form 2 + 3N coupled nonlinear algebraic relations for the 2 + 3N variables *viz.* $P, \rho_1, \epsilon_k, V_k, U_k$. The calculations start with a known initial distribution of the above variables. At each computational cycle their values are updated by solving the finite difference equations.

THE SOLUTION TECHNIQUE

An iterative technique is used to solve the finite difference equations given in the previous section. To facilitate the particular method of solution the equations are recast in the following form. The momentum equations could be collected together in a matrix form. Thus equation (8.18) gives

$$\begin{pmatrix} \delta t\sum_{l=1}^{N} K_{1l} + \epsilon_1\rho_1 & -\delta t K_{12} & \dots & -\delta t K_{1N} \\ -\delta t K_{21} & \delta t\sum_{l=1}^{N} K_{2l} + \epsilon_2\rho_2 & \dots & -\delta t K_{2N} \\ \vdots & \vdots & & \vdots \\ -\delta t K_{N1} & -\delta t K_{N2} & \dots & \delta t\sum_{l=1}^{N} K_{Nl} + \epsilon_N\rho_N \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix} =$$

$$\begin{pmatrix} (\overline{\epsilon_1\rho_1 u_1}) - \frac{\delta t}{\delta r}\epsilon_1\,{}^{n+1}(P_{i+1} - P_i)^j + \frac{\delta t}{\delta r}G_1((\epsilon_1)_{i+1} - (\epsilon_1)_i) \\ (\overline{\epsilon_2\rho_2 u_2}) - \frac{\delta t}{\delta r}\epsilon_2\,{}^{n+1}(P_{i+1} - P_i)^j + \frac{\delta t}{\delta r}G_2((\epsilon_1)_{i+1} - (\epsilon_1)_i) \\ \vdots \\ (\overline{\epsilon_N\rho_N u_N}) - \frac{\delta t}{\delta r}\epsilon_N\,{}^{n+1}(P_{i+1} - P_i)^j + \frac{\delta t}{\delta r}G_N((\epsilon_1)_{i+1} - (\epsilon_1)_i) \end{pmatrix} \tag{8.24}$$

and equation (8.19) gives

$$\begin{pmatrix} \delta t\sum_{l=1}^{N} K_{1l} + \epsilon_1\rho_1 & -\delta t K_{12} & \dots & -\delta t K_{1N} \\ -\delta t K_{21} & \delta t\sum_{l=1}^{N} K_{2l} + \epsilon_2\rho_2 & \dots & -\delta t K_{2N} \\ \vdots & \vdots & & \vdots \\ -\delta t K_{N1} & -\delta t K_{N2} & \dots & \delta t\sum_{l=1}^{N} K_{Nl} + \epsilon_N\rho_N \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix} =$$

$$\begin{pmatrix} (\overline{\epsilon_1\rho_1 v_1}) - \frac{\delta t}{\delta z}\epsilon_1 \; {}^{n+1}\left(P^{j+1} - P^j\right)^j + \frac{\delta t}{\delta z}G_1\left((\epsilon_1)^{j+1} - (\epsilon_1)^j\right) \\ (\overline{\epsilon_2\rho_2 v_2}) - \frac{\delta t}{\delta z}\epsilon_2 \; {}^{n+1}\left(P^{j+1} - P^j\right)^j + \frac{\delta t}{\delta z}G_2\left((\epsilon_1)^{j+1} - (\epsilon_1)^j\right) \\ \vdots \\ (\overline{\epsilon_N\rho_N v_N}) - \frac{\delta t}{\delta z}\epsilon_N \; {}^{n+1}\left(P^{j+1} - P^j\right)^j + \frac{\delta t}{\delta z}G_N\left((\epsilon_1)^{j+1} - (\epsilon_1)^j\right) \end{pmatrix} \tag{8.25}$$

The gas continuity equation (Equation (8.15) for $k = 1$) can be written as

$$\begin{aligned} D_i^j &= -{}^{n+1}(\epsilon_1\rho_1)_i^j + {}^{n}(\epsilon_1\rho_1)_i^j - \frac{\delta t}{r_i\delta r}\,{}^{n+1}\langle(\epsilon_1\rho_1 r)u_1\rangle_i^j \\ &\quad - \frac{\delta t}{\delta z}\,{}^{n+1}\langle(\epsilon_1\rho_1)v_1\rangle_i^j \end{aligned} \tag{8.26}$$

where D_i^j is the residue of the gas continuity equation, which should be equal to zero for absolute convergence. To solve the finite difference equations the pressure in each computational cell is corrected iteratively such that D_i^j meets a convergence criterion. The iterations are continued till the convergence criterion is met in all the computational cells simultaneously. The details of this iterative procedure are given below.

1. Calculate the *tilde* quantities using equations (8.20) and (8.21).
2. Calculate the drag coefficients using equations (8.5) through (8.14).
3. Estimate the new time level velocities solving equations (8.24) and (8.25).
4. Calculate β_i^j, approximately given by

$$\frac{1}{\beta_i^j} = \frac{\partial D_i^j}{\partial P_i^j} = \frac{\epsilon_1}{(C_i^j)^2} + \frac{(\delta t)^2}{r_i(\delta r)^2}\left(r_{i+\frac{1}{2}}(\epsilon_1)_{i+\frac{1}{2}}^j + r_{i-\frac{1}{2}}(\epsilon_1)_{i-\frac{1}{2}}^j\right) + \frac{(\delta t)^2}{(\delta z)^2}\left((\epsilon_1)_i^{j+\frac{1}{2}} + (\epsilon_1)_i^{j-\frac{1}{2}}\right) \tag{8.27}$$

5. Correct the pressure iteratively. Refer to Figure 8.3 which gives the details of this procedure in terms of FORTRAN variables.The computations start with the left–bottom corner fluid cell. Pressure corrections are done in a cell till convergence is obtained or the number of iterations exceed an inner iterations limit. Then the computations proceed from left to right and from bottom to top till the entire computational regime is covered. At the end of such a computational sweep, if convergence was not obtained in any of the cells, the sweeps are repeated. The number of such sweeps are restricted by an outer iterations limit. The iterative procedure for a single cell involves the following steps.

a. Calculate D_i^j using equation (8.24). If $|D_i^j| < \delta_g$ go to step **e**

b. If $|D_i^j| > \delta_g$ the pressure in the cell needs to be adjusted. Initial adjustments are done using Newtons method,

$$^{m+1}P_i^j = {}^{m}P_i^j - \omega\, \beta_i^j\, D_i^j$$

where ω is a relaxation parameter near unity. Newton's method is continued till D_i^j changes sign. After D_i^j changes sign the next pressure correction is done using a secant method (Refer to Figure 8.4a), where

$$\beta = \frac{P_1 - P_2}{D_1 - D_2}$$

There after the pressure corrections are done using a constrained two sided secant

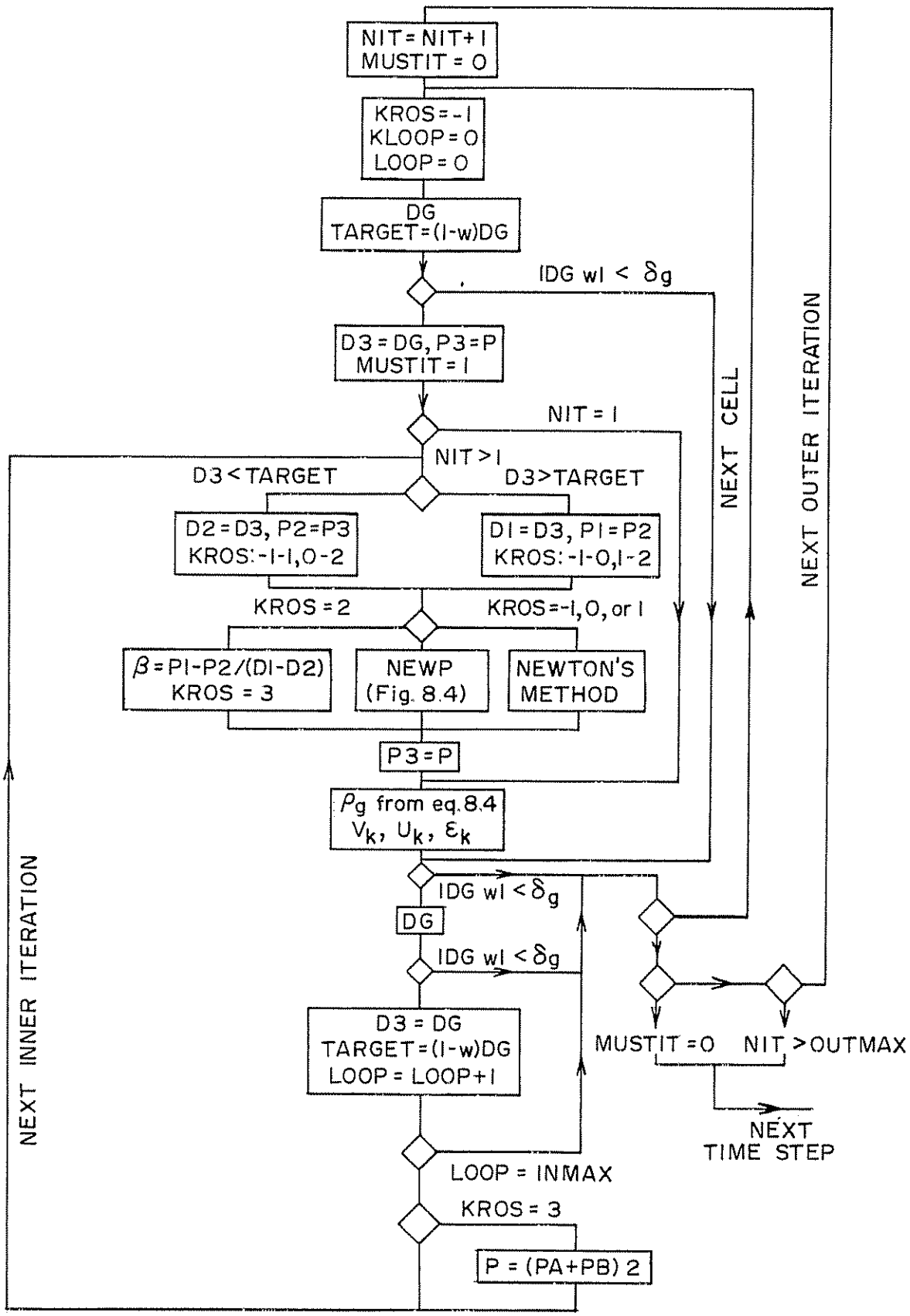
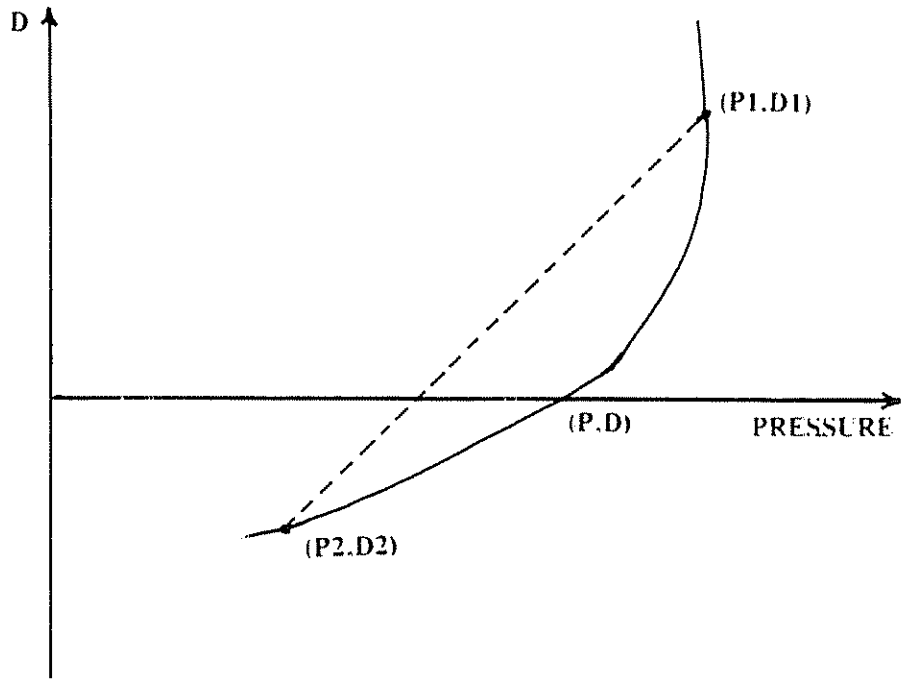


Figure 8.3. The Iteration Loop



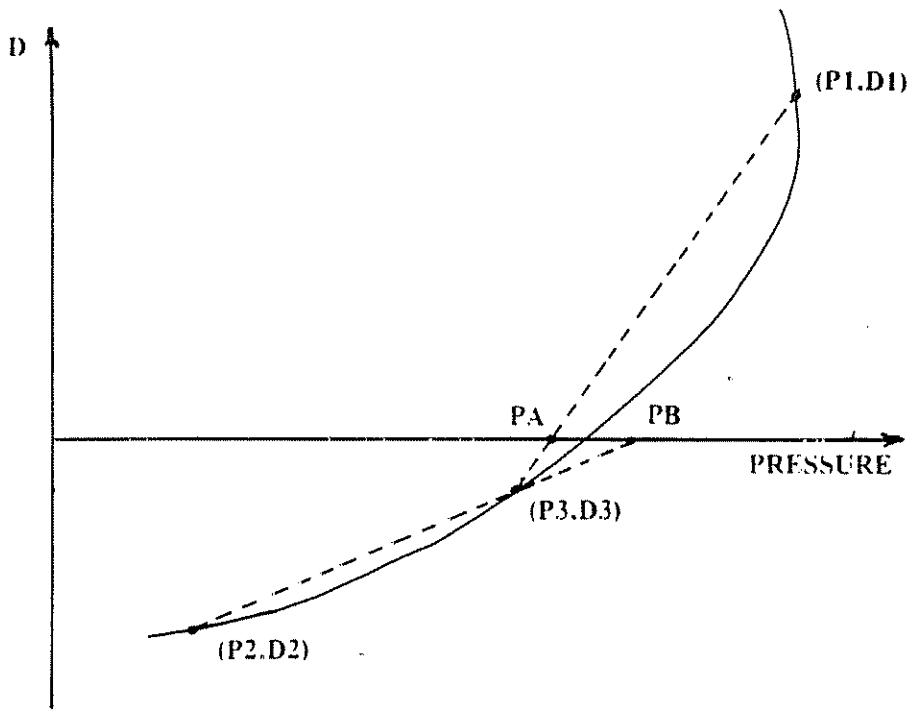


Figure 8.4. The Secant Methods for Pressure Correction

method as illustrated in Figure 8.4b. Given the three points 1, 2, and 3 of which 1 and 2 bracket the desired pressure and 3 lies between them, the pressures P_A and P_B are determined by straight line extrapolation and interpolation, respectively. The new estimate of the advanced time pressure is then computed as ${}^{n+1}P_i^j = \frac{1}{2}(P_A + P_B)$. If the pressure P_A should lie outside the interval P_1 to P_3, it is given the value $\frac{1}{2}(P_1 + P_3)$. After ${}^{n+1}P_i^j$ is estimated, point 2 is discarded and points 1 and 3 are retained as improved bounds for the next pressure estimate.

c. Calculate the velocities using equations (8.24) and (8.25).

d. Calculate the liquid mass fluxes.

e. Solve the solids continuity equations for ϵ_k.

f. Using equation (8.3) find the new value of ϵ_1.

g. If $|D_i^j| > \delta_g$ go to step **a.**

At the beginning of the simulation the distribution of the field variables is given by the initial conditions. Uniform initial conditions can be specified using the input data. Non-uniform initial conditions should be programmed.

Several boundary conditions around the computing mesh perimeter are programmed in the code:

No-slip rigid walls,

Free-slip rigid walls,

Prescribed inflow,

Continuative outflow,

Prescribed outflow pressure.

The rigid walls can be specified along any mesh boundary. The free–slip wall can serve as a reflective plane of symmetry. Two inflow opening each can be specified along the bottom and left boundaries. Similarly two outflow openings each can be prescribed along the top and right boundaries.

The boundary conditions are enforced using the fictitious boundary cells

shown in Figure 8.1. For rigid walls the tangential velocities in the fictitious cells are set by reflection for a free–slip wall and by reflection with a change in sign for no–slip walls. The continuative outflow boundaries are gradient free in their normal direction.

Flow around rectangular obstacles may be calculated by specifying rectangular obstacles in the computing mesh interior. Obstacles are built from rectangular cells. Their boundaries can be a combination of free–slip and no–slip rigid walls. The obstacle boundary conditions are enforced in the same way as they are around the mesh perimeter. To enforce the velocity boundary condition properly, obstacles must be at least two cells wide.

The fluid cells, the boundary cells, and the obstacle cells are distinguished by the program using cell flags. The different cell flags and the corresponding cell types are given below.

Flag	Cell Type
1	Fluid cell
2	Solid cell with free-slip boundaries
3	Solid cell with no-slip boundaries
4	Continuous outflow boundary cell
5	Specified influx inflow cell
6	Specified inflow pressure
7	Specified outflow pressure

PROGRAM ORGANIZATION

A flow chart of the program with the various subroutines is given in Figure 8.5. An alphabetical listing of the various subroutines and their functions are as follows:

BDRY	Set velocity boundary conditions. Reflect cell center quantities.
BETAS	Calculate β
CNVERT	Calculate initial values of microscopic and macroscopic densities.
EOSG	Calculate the gas density and the speed of sound in gas.
FLIC	Sets cell flags based on input data.
INDEX	Calculate indices for array quantities.
INTER	Calculate particle–particle friction coefficient.
ITER	Perform the iterative solution of the difference equations.
KDRAGG	Gas–solids drag for low solids concentration.
KDRAGL	Gas–solids drag for high solids concentration.
KDRAGS	Select KDRAGG or KDRAGL.
MASFG	Calculates mass fluxes for the gas.
MASFK	Calculates mass fluxes for the solids.
MATS	Calculates the matrix components for velocity calculation.
NEWP	Calculates the new estimate of advanced time pressure from three (P,D) points.
PROG	Control the program flow and outputting.
SETC	Sets the C array used to store necessary constants.
SETUP	Initialize the radii and all the field variables.
STXTRA	Initialize constant pressure out flow boundary cells.
TAPERD	Read the restart file.

TAPEWR	Write the restart file.
TILDE	Calculate the *tilde* quantities.
UGMOMF	Calculate the radial momentum flux of gas.
ULMOMF	Calculate the radial momentum flux of solids.
VELSK	Calculate the velocities at the cell boundaries.
VGMOMF	Calculate the axial momentum flux of gas.
VLMOMF	Calculate the axial momentum flux of solids.

INPUT DATA

1. NAME (format 40A2) Problem identifier

2. IB2,JB2,NSOLID (format free); IB2 - the number of cells in the radial direction + two fictitious cells; JB2 - the number of cells in the axial direction + two fictitious cells; NSOLID - the number of particulate phases.

3. (SCALE(I), I=1,4) (format free); SCALE(1) - the length scale; SCALE(2) - the velocity scale; SCALE(3) - the density scale; SCALE(4) - the temperature scale.

4. ITC,DR,DZ,JH (format free); ITC = 0 for rectangular coordinates, = 1 for cylindrical coordinates, = 2 for one dimensional spherical coordinates; DR - δr; DZ - δz; JH - no of rows from the bottom in the axial direction in which solids are initially present.

5. (KFLIN(M), M=1,4) (format free); Cell flags for the inflow openings; KFLIN(1) - Bottom–Left; KFLIN(2) - Bottom–Right; KFLIN(3) - Left–Bottom; KFLIN(4) - Left–Top.

6. (FLO(M), M=1,8) (format free); Coordinates for the inflow openings; FLO(1), FLO(2) - r coordinate for Bottom–Left opening; FLO(3), FLO(4) - r coordinate for the Bottom–Right opening; FLO(5), FLO(6) - z coordinate for the Left–Bottom opening; FLO(7), FLO(8) - z coordinate for the Left–Top opening.

7. (KFLOUT(M), M=1,4) (format free); Cell flags for outflow openings; KFLOUT(1) - Top–Left; KFLOUT(2) - Top–Right; KFLOUT(3) - Right–Top; KFLOUT(4)

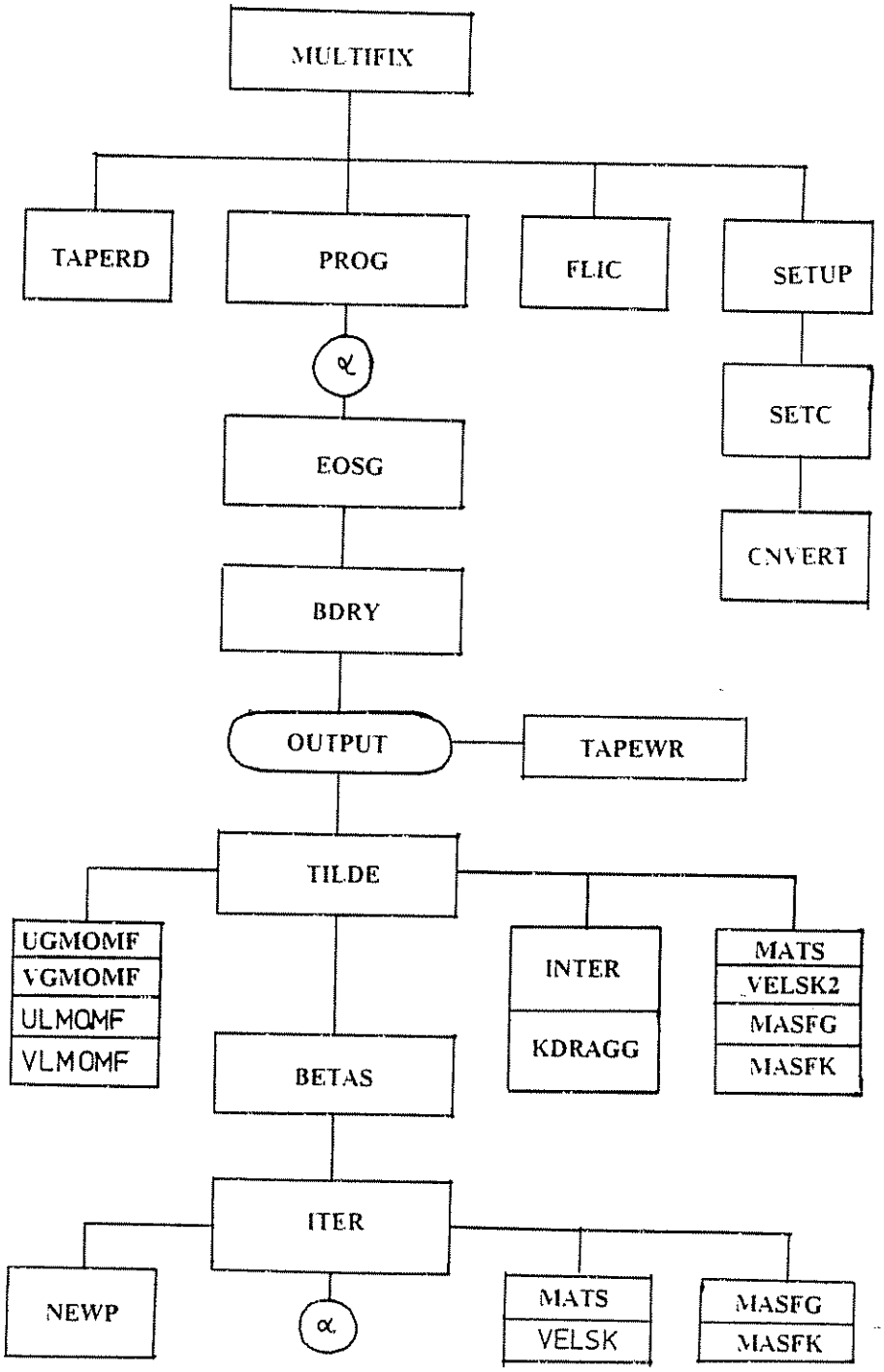


Figure 8.5. Program Flow Chart

- Right–Bottom.

8. (FLO(M), M=9,16) (format free); Coordinates for the outflow openings; FLO(9), FLO(10) - r coordinate for the Top–Left opening; FLO(11), FLO(12) - r coordinate for the Top–Right opening; FLO(13), FLO(14) - z coordinate for the Right–Top opening; FLO(15), FLO(16) - z coordinate for the Right–Bottom opening.

9. (NSL(M), M=1,4) (format free); Indicates free–slip or no–slip boundary condition for rigid walls around the computing mesh perimeter. Values of 0 for free–slip or 1 for no–slip are assigned for Bottom, Left, Top, and Right boundaries in that order. These values are ignored across inflow and out flow openings.

10. NO (format free); number of obstacles.

11. (NSO(N), (OB(M,N), M=1,4), N=1,NO) (format free); NSO(N) = 0 free–slip for obstacle N or 1 no–slip for obstacle N; Obstacle N is defined by the coordinates, OB(1,N) - r coordinate of left side, OB(2,N) - r coordinate of right side, OB(3,N) - z coordinate of bottom, and OB(4,N) - z coordinate of top.

12. (DK(K), K=1,NSOLID) (format free); Characteristic diameters of the particulate phases.

13. (RK(K), K=1,NSOLID) (format free); Microscopic densities of the particulate phases.

14. UO,VO,PO,EPO,TEMPO,GRAV (format free); UO - initial radial gas velocity; VO - initial axial gas velocity; PO - initial pressure; EPO - initial void fraction ; TEMPO - initial temperature; GRAV - gravitational accleration in z–direction; Non-uniform initial data may be specified by modifying the subroutine SETUP.

15. In flow data in the order given below (format free)

	Gas	Velocity			
	radial	axial	pressure	ϵ_1	temperature
Bottom-Left	UINL	VINL	PINL	EPINL	TEMPINL
Bottom-Right	UINR	VINR	PINR	EPINR	TEMPINR

Left-Bottom	UINB	VINB	PINB	EPINB	TEMPINB
Left-Top	UINT	VINT	PINT	EPINT	TEMPINT

16. ITD (format free); ITD $<$ 2 no restart; ITD $\geq$ 2 read the initial data from a restart file.

17. TIME,TSTOP,DT,CYCLE (format free); TIME - initial time of the problem; TSTOP - the time at which the calculation is to stop; DT - the time step δt; CYCLE - the number of the computational cycle at which the calculation is to begin.

18. TPR,TDUMP (format free); TPR - the time interval for storing data on a disk file which may be printed; TDUMP - the time interval for binary output to a disk file which may be used as a restart file.

SETTING UP A PROBLEM

First of all choose an appropriate coordinate system from rectangular, cylindrical or one dimensional spherical coordinates. Use the symmetries of the problem to reduce the computational region. Specify free–slip walls along the lines of symmetry. Fix the coordinate system such that the inflow is from the computing mesh bottom or left and the outflow is from the top or right. The outflow boundary should be located in such a way that the flow is normal to the outflow boundary.

Program non–uniform initial conditions, if any, by modifying the subroutine SETUP.

Specify δr and δz according to the degree of resolution required. Sometimes these dimensions are governed by the sizes of inflow or outflow openings. Discretize the computational regime using integral number of divisions in r and z directions. The total number of computational cells are typically 100 - 1500.

The time step, δt, is limited by the fluid transit time through a cell. Thus

$$\delta t \quad < \quad \frac{\min.(\delta r,\ \delta z)}{\text{max. velocity}}$$

Note that there is no sonic time step limitation, since the method is a generalization of the ICE technique. However, in some cases the time step required for proper convergence could be much less than what is given by the above expression.

A List of FORTRAN Symbols

In the following list of FORTRAN symbols the array quantities are indicated by an index. They are all part of a common block. The array dimensions of most of them need to be changed when the input data file is changed. The indices shown in the following table and the respective array dimensions to be used are: I -> IB2 ; IJ -> IB2 * JB2 ; K -> Number of particulate phases ; N -> Total number of phases ; the rest are left unchanged unless othewise specified.

FORTRAN Symbol	Definition
ABETA(IJ)	$\frac{1}{\beta_i^j}$ Computed in BETAS and ITER.
APPU(M, IJ)	r-component of $-(K_{kl})_i^j \delta t$ as off–diagonal elements and $\delta t \sum_{l=1}^{N} (K_{kl})_i^j$ as diagonal elements. M -> N*(N+1)/2 where N is the total number of phases
APPV(M, IJ)	z-component of $-(K_{kl})_i^j \delta t$ as off–diagonal elements and $\delta t \sum_{l=1}^{N} (K_{kl})_i^j$ as diagonal elements. M -> N*(N+1)/2 where N is the total number of phases
AU(N, N)	Components of the matrix in equation (24) for the location $i + \frac{1}{2}, j$
AU1(N, N)	Components of the matrix in equation (24) for the location $i - \frac{1}{2}, j$
AV(N, N)	Components of the matrix in equation (25) for the location $i, j + \frac{1}{2}$
AV1(N, N)	Components of the matrix in equation (25) for the location $i, j - \frac{1}{2}$

BU(N)	The vector in equation (24) for the location $i+\frac{1}{2}, j$
BU1(N)	The vector in equation (24) for the location $i-\frac{1}{2}, j$
BV(N)	The vector in equation (25) for the location $i, j+\frac{1}{2}$
BV1(N)	The vector in equation (25) for the location $i, j-\frac{1}{2}$
C(100)	Storage for constants initialized in SETC.
CONV(IJ)	δg - Convergence criteria computed in BETAS
CYCLE	n - The calculation cycle
DG	D_i^j - Defined in equation (26)
DK(K)	d_k - Characteristic diameter of phase k
DKF(K, K)	$\frac{\delta t \rho_k \rho_l (d_k+d_l)^2}{2\left(\rho_k d_k^3+\rho_l d_l^3\right)}$
DR	δr
DT	δt
DTODR	$\frac{\delta t}{\delta r}$
DTODZ	$\frac{\delta t}{\delta z}$
DTORBDR(I)	$\frac{\delta t}{r_{i+\frac{1}{2}} \delta r}$
DTORDR(I)	$\frac{\delta t}{r_i \delta r}$
DZ	δz
D1, D2, D3	Values of D_i^j (See Figure 8.4)
EP(IJ)	$(\epsilon_1)_i^j$
EPINB	Input data line no. 15
EPINL	Input data line no. 15
EPINR	Input data line no. 15

EPINT	Input data line no. 15
EPN(IJ)	${}^{n}(\epsilon_1)_i^j$
EPO	Input data line no. 14
EPSL(K, K)	$[(\Phi_k - \Phi_l) + (1-a)(1-\Phi_k)\Phi_l][\Phi_k + (1-\Phi_k)\Phi_l]$
EPSTAR	ϵ^*
EPSU(K, K)	$(1-a)[\Phi_k + (1-\Phi_k)\Phi_l]$
FL(IJ)	Cell flags set up in FLIC
FLO(M)	Input data line no. 6 and 8
GEP(100)	Table of solids stress for explicit treatment if desired
GRAV	g - Input data line no. 14
I	i - Computing mesh column index
IB	Number cells in the radial direction excluding the two fictitious columns; IB = IB2-2
IB1	IB1 = IB2 - 1
IB2	Total number of cells in the radial direction
IJ	Index of quantities for cell i,j; IJ = I + (J-1)*IB2
IJB	Index of cell centered quantities associated with cell i,j-1; see PROG
IJBR	Index of cell centered quantities associated with cell i+1, j-1; see INDEX
IJL	Index of cell centered quantities associated with cell i-1, j; see PROG
IJM	Index of cell i, j-1
IJP	Index of cell i, j+1
IJR	Index of cell centered quantities associated with cell i+1, j; see PROG

IJRR	Index of cell centered quantities associated with cell i+2, j; see INDEX
IJT	Index of cell centered quantities associated with cell i, j+1; see PROG
IJTL	Index of cell centered quantities associated with cell i-1, j+1; see INDEX
IJTR	Index of cell centered quantities associated with cell i+1, j+1; see PROG
IJTT	Index of cell centered quantities associated with cell i, j+2; see INDEX
IMJ	Index of cell i-1, j
IMJM	Index of cell i-1,j-1
IMJP	Index of cell i-1, j+1
INDS(5, IJ)	Storage of IJR, IJL, IJT, IJB, and IJTR for each i,j
IPJ	Index of cell i+1, j
IPJM	Index of cell i+1, j-1
IPJP	Index of cell i+1, j+1
ITC	Input data line no. 4
ITD	Input data line no. 16
J	j - Computing mesh row index(z - direction)
JB	Number of cells in the axial direction excluding the two fictitious rows; JB = JB2-2
JB1	JB1 = JB2 - 1
JB2	Total number of cells in the axial direction
JH	Input data line no. 4
K	Index for the phases

KPGU(K, IJ)	r - component of gas - solids friction
KPGV(K, IJ)	z - component of gas - solids friction
MMUG	Viscosity of gas
NAME(M)	Input data line no. 1
NIT	Iteration counter used in ITER
NO	Input data line no. 10
NPHASE	Total number of phases
NSL(M)	Input data line no. 9
NSO(M)	Input data line no. 11
NSOLID	Number of particulate phases
OB(M, N)	Input data line no. 11
P(IJ)	P_i^j - Pressure in cell i,j
PHI(K)	Φ_k
PHILIM(K, K)	$\frac{\Phi_k}{\Phi_k+(1-\Phi_k)\Phi_l}$
PHIS	ϕ - the shape factor
PINB	Input data line no. 15
PINL	Input data line no. 15
PINR	Input data line no. 15
PINT	Input data line no. 15
PO	Input data line no. 14
P1, P2, P3	Values of P_i^j (Figure 8.4)
R(I)	r_i - Radial coordinate of the center of cell i,j
RAGS	C^{-2}
RB(I)	$r_{i+\frac{1}{2}}$ - Radial coordinate of the right boundary of cell i,j
RDR	$\frac{1}{\delta r}$

RDR2	$\frac{1}{(\delta r)^2}$
RDZ	$\frac{1}{\delta z}$
RDZ2	$\frac{1}{(\delta z)^2}$
RGFR(IJ)	Flux of $\epsilon_1 \rho_1$ across the right boundary of cell i,j
RGFT(IJ)	Flux of $\epsilon_1 \rho_1$ across the top boundary of cell i,j
RGP(IJ)	$(\epsilon_1 \rho_1)_i^j$
RGPN(IJ)	${}^n(\epsilon_1 \rho_1)_i^j$
RL(K)	ρ_k - Solids density
RLFRK(K, IJ)	Flux of $\epsilon_k \rho_k$ across the right boundary of cell i,j
RLFTK(K, IJ)	Flux of $\epsilon_k \rho_k$ across the top boundary of cell i,j
RLIM	Volume fraction limit to determine the complete drainage of a particulate phase from a cell
RLK(K, IJ)	$(\epsilon_k \rho_k)_i^j$
RLKN(K, IJ)	${}^n(\epsilon_k \rho_k)_i^j$
ROG(IJ)	$(\rho_1)_i^j$
RRB(I)	$\frac{1}{r_{i+\frac{1}{2}}}$
RRIDR(I)	$\frac{1}{r_i \delta r}$
RUG(IJ)	$(\overline{\epsilon_1 \rho_1 u_1})$
RUK(K, IJ)	$(\overline{\epsilon_k \rho_k u_k})$
RVG(IJ)	$(\overline{\epsilon_1 \rho_1 v_1})$
RVK(K, IJ)	$(\overline{\epsilon_k \rho_k v_k})$
SCALE(4)	Input data line no. 3
TDUMP	Input data line no. 22

TEMPINB	Input data line no. 15
TEMPINL	Input data line no. 15
TEMPINR	Input data line no. 15
TEMPINT	Input data line no. 15
TEMPO	Input data line no. 14
TG	Gas temperature
TIME	Input data line no. 17
TL	Solids temperature
TPR	Input data line no. 22
TSTOP	Input data line no. 17
UG(IJ)	$(u_1)^j_{i+\frac{1}{2}}$
UGFB(I)	Radial momentum flux for the gas across the bottom boundary of the momentum control volume centered about the point $i + \frac{1}{2}, j$
UGFL	Radial momentum flux for the gas across the left boundary of the momentum control volume described above
UGFR	Radial momentum flux for the gas across the right boundary of the momentum control volume described above
UGFT	Radial momentum flux for the gas across the top boundary of the momentum control volume described above
UINB	Input data line no. 15
UINL	Input data line no. 15
UINR	Input data line no. 15
UINT	Input data line no. 15

UK(K, IJ)	$(u_k)^j_{i+\frac{1}{2}}$
ULFB(K, I)	Radial momentum flux for the solids across the bottom boundary of the momentum control volume centered about the point $i+\frac{1}{2}, j$
ULFL(K)	Radial momentum flux for the solids across the left boundary of the momentum control volume described above
ULFR	Radial momentum flux for the solids across the right boundary of the momentum control volume described above
ULFT	Radial momentum flux for the solids across the top boundary of the momentum control volume described above
UO	Input data line no. 14
VG(IJ)	$(v_1)^{j+\frac{1}{2}}_i$
VGFB(I)	Axial momentum flux for the gas across the bottom boundary of the momentum control volume centered about the point $i, j+\frac{1}{2}$
VGFL	Axial momentum flux for the gas across the left boundary of the momentum control volume described above
VGFR	Axial momentum flux for the gas across the right boundary of the momentum control volume described above
VGFT	Axial momentum flux for the gas across the top boundary of the momentum control volume described above

VINB	Input data card no. 15
VINL	Input data card no. 15
VINR	Input data card no. 15
VINT	Input data card no. 15
VK(K, IJ)	$(v_k)_i^{j+\frac{1}{2}}$
VLFB(K, IJ)	Axial momentum flux for the solids across the bottom boundary of the momentum control volume centered about the point $i, j + \frac{1}{2}$
VLFL(K)	Axial momentum flux for the solids across the left boundary of the momentum control volume described above
VLFR	Axial momentum flux for the solids across the right boundary of the momentum control volume described above
VLFT	Axial momentum flux for the solids across the top boundary of the momentum control volume described above
VO	Input data line no. 14

```
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER FL,CYCLE
      REAL*8 KPGU,KPGV,LEFT,MMUG
C      COMMON BLOCKS
      COMMON / PARAM1 /
     1 MMUG,                 DG,                 DR,        DT,
     1 DTODR,     DTODZ,     DZ,        D1,        D2,        D3,
     1 EPSTAR,       GRAV,
     1      PINB,      PINL,      PINR,      PINT,      PO,
     1 P1,        P2,        P3,        PHIS,      RAGS,
     1 RDR,       RDR2,      RDZ,       RDZ2,      RLIM
      COMMON / PARAM2 /
     1 TARGET,          TEMINB ,   TEMINL,   TEMINR,   TEMINT,
     1 TEMPO,  EPINB,     EPINL,
     1 EPINR,     EPINT,     EPO,           TIME,    TDUMP,
     1       TPR,       TSTOP,     UGFL,      UGFR,      UGFT,
     1 UINB,      UINL,      UINR,      UINT,                 ULFR,
     1 ULFT,      UO,        VGFL,      VGFR,      VGFT,
     1 VLFR,      VLFT,      VINB,      VINL,      VINR,      VINT,
     1 VO,
     1     TG,      TL
      COMMON / PARAM3 /
     1 C(100),    FLO(16),
     1 OB(4,37),  SCALE(4),          DK(  2),GEP(100),DKF(  2,  2),
     1 RL(  2), PHI(  2),PHILIM(  2,  2),EPSL(  2,  2),EPSU(  2,  2)
      COMMON / PARAMI /
     1 I,IB,IB1,NO,
     1 IB2,       IJ,        IJB ,      IJBR,      IJL,       IJM,
     1 IJP,       IJR,       IJRR,      IJT,       IJTL,      IJTR,
     1 IJTT,      IMJ,       IMJM,      IMJP,      IPJ,       IPJM,
     1 IPJP,      ITC,       ITD,       J ,        JB,        JB1,
     1 JB2,               K,           NPHASE,    CYCLE,
     1                NIT,            NSOLID,
     1  NAME(40),NSL(4),NSO(37),FL(  54),INDS(  54,5)
      COMMON
     1 ABETA(  54 ),CONV(  54 ),
     1P    (  54  ), RGFR (  54  ), RGFT (  54  ),
     1RGP  (  54  ), RGPN (  54  ),ROG(  54  ),
     1RUG  (  54  ), RVG  (  54  ),
     1EP   (  54  ), EPN  (  54  ),
     1APPU(  6,  54),APPV(  6,  54),BU1(  3),BV1(  3),BU(  3),
     1BV(  3),AU1(  3,  3),AV1(  3,  3),AU(  3,  3),AV(  3,  3)
      COMMON
     1                RLFRK(  2,  54),RLFTK(  2,  54),
     1 KPGU(  2,  54),KPGV(  2,  54),
     1   RUK(  2,  54),RVK(  2,  54),
     1RLK(  2,  54), RLKN(  2,  54),
     1UG   (  54  ), UK(  2,  54), VK(  2,  54),
     1 VG  (  54  )
      COMMON
     1DTOBDR  ( 3), DTORDR  ( 3),           R      ( 3),
     1RB      ( 3), RRB     ( 3), RRIDR   ( 3),
```

```
1 UGFB     ( 3), ULFB     (  2, 3),
1VGFB    ( 3), VLFB     (  2, 3),
1ULFL   (  2), VLFL   (  2)
```

```
C
C*******************************************************************
C          Transient Two-Dimensional Multiphase Flow Program
C          ..................................................
C                          Developed by
C                           M. SYAMLAL
C                Illinois Institute of Technology
C                       Chicago, Il 60616
C                           May 1985
C
C*******************************************************************
C
      INCLUDE 'GCOME.FOR'
      COMMON/KFLIO/KFLIN(4),KFLOUT(4)
C
C          Input Data
C
      READ(5,100) NAME
      READ(5,*) IB2,JB2,NSOLID
      READ (5,*) (SCALE(I),I=1,4)
      READ(5,*) ITC,DR,DZ,JH
      READ(5,*) (KFLIN(M),M=1,4)
      READ(5,*) (FLO(M),M=1,8)
      READ(5,*) (KFLOUT(M),M=1,4)
      READ(5,*) (FLO(M),M=9,16)
      READ(5,*)(NSL(M),M=1,4)
      READ(5,*)NO
      IF(NO.EQ.0)GOTO 25
      DO 20 N=1,NO
20    READ(5,*)NSO(N),(OB(M,N),M=1,4)
25    READ(5,*)(DK(K),K=1,NSOLID)
      READ(5,*)(RL(K),K=1,NSOLID)
      READ(5,*)UO,VO,PO,EPO,TEMPO,GRAV
      READ(5,*) UINL,VINL,PINL,EPINL,TEMINL,UINR,VINR,PINR,EPINR,
     1          TEMINR,UINB,VINB,PINB,EPINB,TEMINB ,UINT,VINT,PINT,
     2          EPINT,TEMINT
      READ(5,*) ITD
      READ (5,*) TIME,TSTOP,DT,CYCLE
      READ (5,*) TPR,TDUMP
C
C-------------------------------------------------------------------
C
C
C----------Read Restart File if necessary
C
      IF (ITD.GE.2) CALL TAPERD
C
C-------------------------------------------------------------------
C
C
C---------Echo the input data
```

```
C
      WRITE(6,200) NAME
      WRITE(6,210) (SCALE(I),I=1,4)
      WRITE(6,220) ITC,IB2,JB2,DR,DZ,JH
      WRITE(6,230)(KFLIN(I),I=1,4),(FLO(M),M=1,8),(KFLOUT(I),I=1,4),
     1(FLO(M),M=9,16)
      WRITE(6,240) (NSL(M),M=1,4)
      WRITE(6,250) NO
      IF(NO.EQ.0) GO TO 50
      WRITE(6,252)
      DO 40 N=1,NO
   40 WRITE(6,254) NSO(N),(OB(M,N),M=1,4)
   50 WRITE(6,251) GRAV
      WRITE(6,260) UO,VO,PO,EPO,TEMPO
      WRITE(6,270) UINL,VINL,PINL,EPINL,TEMINL,UINR,VINR,PINR,
     1                 EPINR,TEMINR,UINB,VINB,PINB,EPINB,TEMINB ,
     2                 UINT,VINT,PINT,EPINT,TEMINT
      WRITE(6,274)
      WRITE(6,275)(DK(I),RL(I),I=1,NSOLID)
      WRITE(6,280) ITD
      WRITE(6,290) TIME,TSTOP,DT,CYCLE
      WRITE(6,300) TPR,TDUMP
C
C-----------------------------------------------------------------------
C
      REWIND(9)
      NPHASE=NSOLID+1
      IB=IB2-2
      IB1=IB2-1
      JB=JB2-2
      JB1=JB2-1
      TG=TEMPO
      TL=TEMPO
C
C--------Initialize cell flags, constants, and dependant variables
C
      CALL FLIC
      CALL SETUP (JH)
C
C-----------------------------------------------------------------------
C
C    ******************************************************************
C    --------------------- MARCH IN TIME ----------------------------
      CALL PROG
C    ******************************************************************
      STOP
  100 FORMAT(40A2)
  200 FORMAT(1X,'MULTIFIX   PROBLEM IDENTIFIER -  ',40A2)
  210 FORMAT(/,1X,'SCALING    LENGTH(CM)=',1PE8.2,  '  VELOCITY',
     1 '(CM/SEC)=',1PE 8.2, '  DENSITY(GM/CC)=',1PE 8.2,    '  TEMPER'
     2,'ATURE(DEG.K)=',1PE 8.2  )
  220 FORMAT(/,1X,'GEOMETRY',/,'      1. COORDINATES (RECT=0, CYLIND'
     1,'=1,SPHER=2) =',I3,/,
```

```
     1   '       2. MESH SIZE,     IB2=',I3,14X,'JB2=',I3,/,'       3. '
     2 ,'CELL SIZE,      DR=',1PE11.4,'   DZ=',1PE11.4,/,'       4. TOP'
     3, 'OF BED    JH=',I12)
  230 FORMAT(/,1X,'      5. INFLOW OPENINGS       CELL FLAGS     BO'
     1,'TTOM-L=',I3,11H  BOTTOM-R=,I3,9H  LEFT-B=,I3,9H  LEFT-T=,I3/,
     212X,9HA. BOTTOM,10X,1P4E11.4/,12X,7HB. LEFT,12X,1P4E11.4/
     351H      6. OUTFLOW OPENINGS     CELL FLAGS     TOP-L=,I3,
     48H  TOP-R=,I3,10H  RIGHT-B=,I3,10H  RIGHT-T=,I3/,12X,
     56HA. TOP,13X,1P4E11.4/,12X,8HB. RIGHT,11X,1P4E11.4)
  240 FORMAT(/,1X,'      7. BOUNDARIES, (FREE-SLIP=0   NO-SLIP=1)'/,14X
     1,'BOTTOM=',I3,'   LEFT=',I3,'   TOP=',I3,'  RIGHT=',I3  )
  250 FORMAT(/,1X,'      8. OBSTACLES,     NO=',I3 )
  251 FORMAT(/,1X,'      9. GRAVITY,       GRAV=',1PE15.7)
  252 FORMAT(/,1X,'             SLIP',22X,'------COORDINATES------')
  254 FORMAT(/,1X,12X,I3,6X,1P4E14.4 )
  260 FORMAT(/,1X,'INITIAL DATA    GAS AND LIQUID '/        '
     1UO=',1PE11.4,'    VO=',1PE11.4,'    PO=',1PE11.4,'    EPO=',1PE
     211.4,'     TO=',1PE11.4)
  270 FORMAT(/,1X,'INFLOW DATA'/'      1. BOTTOM'/'            UNIL='
     1,1PE11.4,'  VINL=',1PE11.4,'  PINL=',1PE11.4,'  EPINL=',1PE11.4
     2,'TEMINL=',1PE11.4/'            UINR=',1PE11.4,'  VINR=',1PE11.
     34,'PINR=',1PE11.4,'  EPINR=',1PE11.4, ' TEMINR=',1PE11.4/'
     4 2.LEFT'/,12X,'UINB=',1PE11.4,'  VINB=',1PE11.4,'  PINB=',1PE11
     5.4,'  EPINB=',1PE11.4,' TEMINB =',1PE11.4/'            UINT=',1
     6PE11.4,'  VINT=',1PE11.4,'  PINT=',1PE11.4,'  EPINT=',1PE11.4,'
     7 TEMINT=',1PE11.4)
274   FORMAT(/,1X,/,' PARTICULATE PHASE DATA',/,'      DIAMETER',7X,
     & 'MICROSCOPIC DENSITY')
275   FORMAT(/,2(6X,G10.3))
  280 FORMAT(/,1X,'CONTROL'/ '       1. DUMP AND RESTART,    ITD=',I3)
  290 FORMAT(/,1X,'      2. TIME AND CYCLE   TSTART=',1PE11.4,'  TST'
     1,'OP=',1PE11.4,'  DT=',1PE11.4,'  CYCLE=',I12)
  300 FORMAT(/,1X,'      3. PRINTING AND DUMPING,     TPR='
     1,1PE11.4,' TDUMP=',1PE11.4)
      END
      SUBROUTINE BDRY
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C   Set velocity boundary conditions. Reflect cell center quantities
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      DO 200 J=2,JB1
      DO 200 I=2,IB1
        IJ=I+(J-1)*IB2
C     Skip if not a fluid cell
        IF(FL(IJ).NE.1) GO TO 200
        CALL INDEX
C     Check cells on right and top
        NFLR=FL(IPJ)
        NFLTR=FL(IPJP)
        NFLT=FL(IJP)
        IF(NFLR.EQ.4.OR.NFLR.EQ.7) GO TO 10
        IF(NFLT.EQ.4.OR.NFLT.EQ.7)GO TO 20
```

```
        GOTO30
C
C            Continuous outflow on the right
C
   10   CONTINUE
        N1=IJ
        N2=IPJ
        IF(NFLR.EQ.7.AND.UG(N1).LE.0.) GO TO 16
      RGFR(N2)=RGFR(N1)
      RGP(N2)=RGP(N1)
      RUG(N2)=RUG(N1)
      EP(N2)=EP(N1)
      DO 15 K=1,NSOLID
      RLFRK(K,N2)=RLFRK(K,N1)
      RLK(K,N2)=RLK(K,N1)
      RUK(K,N2)=RUK(K,N1)
15    CONTINUE
      IF(NFLR.NE.7)P(N2)=P(N1)
      IF(NFLR.EQ.7)THEN
      IJ=N2
      CALL CNVERT
      IJ=N1
      ENDIF
   16   CONTINUE
      IF(NFLTR.GE.4)THEN
      VG(N2)=VG(N1)
      DO 17 K=1,NSOLID
      VK(K,N2)=VK(K,N1)
17    CONTINUE
      ENDIF
      DO 18 K=1,NSOLID
      UK(K,N2)=((R(I+1)+R(I))*RB(I)*UK(K,N1)-R(I+1)*RB(I-1)
     & *UK(K,N1-1))*RRB(I+1)/R(I)
18    CONTINUE
        UG(N2)=((R(I+1)+R(I))*RB(I)*UG(N1)-R(I+1)*RB(I-1)*UG(N1-1))
     1  *RRB(I+1)/R(I)
        IF(NFLT.NE.4.AND.NFLT.NE.7)GO TO 30
C
C     Continuous outflow on the top
C
   20   CONTINUE
        N1=IJ
        N2=IJP
      IF(NFLT.EQ.7.AND.VG(N1).LE.0.)GOTO 26
      RGFT(N2)=RGFT(N1)
      RGP(N2)=RGP(N1)
      RVG(N2)=RVG(N1)
      EP(N2)=EP(N1)
      DO 25 K=1,NSOLID
      RLFTK(K,N2)=RLFTK(K,N1)
      RLK(K,N2)=RLK(K,N1)
      RVK(K,N2)=RVK(K,N1)
25    CONTINUE
```

```
      IF(NFLT.NE.7)P(N2)=P(N1)
      IF(NFLT.EQ.7)THEN
      IJ=N2
      CALL CNVERT
      IJ=N1
      ENDIF
26    CONTINUE
        IF(NFLTR.GE.4) THEN
      UG(N2)=0.0
      DO 27 K=1,NSOLID
      UK(K,N2)=0.0
27    CONTINUE
      ENDIF
        VG(N2)=RGP(N1)*VG(N1)/RGP(N2)
      DO 28 K=1,NSOLID
      VK(K,N2)=0.0
28    CONTINUE
   30   GO TO (60,35,45,60,60,60,60),NFLR
   35   GO TO (60,40,40,60,60,60,60),NFLTR
C
C          Free-slip wall on the right
C
   40   VG(IPJ)=VG(IJ)
      DO 41 K=1,NSOLID
      VK(K,IPJ)=VK(K,IJ)
41    CONTINUE
        GOTO60
   45   GO TO (60,50,50,60,60,60,60),NFLTR
C
C         No-slip wall on the right
C
   50   VG(IPJ)=-VG(IJ)
      DO 51 K=1,NSOLID
      VK(K,IPJ)=-VK(K,IJ)
51    CONTINUE
   60   GO TO (90,65,75,90,90,90,90),NFLT
   65   GO TO (90,70,70,90,90,90,90),NFLTR
C
C        Free-slip wall above
C
   70   UG(IJP)=UG(IJ)
      DO 71 K=1,NSOLID
      UK(K,IJP)=UK(K,IJ)
71    CONTINUE
        GOTO90
   75   GO TO (90,80,80,90,90,90,90),NFLTR
C
C       No-slip wall above
C
   80   UG(IJP)=-UG(IJ)
      DO 81 K=1,NSOLID
      UK(K,IJP)=-UK(K,IJ)
81    CONTINUE
```

```
   90   CONTINUE
        NFLL=FL(IMJ)
        NFLTL=FL(IMJP)
        GO TO (130,105,115,130,130,130,130), NFLL
  105   GO TO (130,110,110,130,130,130,130), NFLTL
C
C        Free-slip wall on the left
C
  110   VG(IMJ)=VG(IJ)
      DO 111 K=1,NSOLID
      VK(K,IMJ)=VK(K,IJ)
111   CONTINUE
        GO TO 130
115     GO TO (130,120,120,130,130,130,130), NFLTL
C
C       No-slip wall on the left
C
  120   VG(IMJ)=-VG(IJ)
      DO 121 K=1,NSOLID
      VK(K,IMJ)=-VK(K,IJ)
121   CONTINUE
  130   NFLB=FL(IJM)
        NFLBR=FL(IPJM)
        GO TO (200,135,145,200,200,200,200),NFLB
  135   GO TO (200,140,140,200,200,200,200),NFLBR
C
C       Free-slip wall below
C
  140   UG(IJM)=UG(IJ)
      DO 141 K=1,NSOLID
      UK(K,IJM)=UK(K,IJ)
141   CONTINUE
        GO TO 200
  145   GO TO (200,150,150,200,200,200,200),NFLBR
C
C       No-slip wall below
C
  150   UG(IJM)=-UG(IJ)
      DO 151 K=1,NSOLID
      UK(K,IJM)=-UK(K,IJ)
151   CONTINUE
  200 CONTINUE
      RETURN
      END
      SUBROUTINE BETAS
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C      Calculate beta for the iteration
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      PARAMETER (DELG = 1.0E-4)
      DO 10 J=2,JB1
      DO 10 I=2,IB1
        IJ=I+(J-1)*IB2
```

```
        IF(FL(IJ).NE.1) GO TO 10
        CALL INDEXA
        RAGS=ROG(IJ)/P(IJ)
        RIGHT=1.
        LEFT=1.
        TOP=1.
        BOT=1.
        NFLR=FL(IPJ)
        NFLL=FL(IMJ)
        NFLT=FL(IJP)
        NFLB=FL(IJM)
        GO TO (2,1,1,2,1,1,2),NFLR
    1   RIGHT=0.
    2   GO TO (4,3,3,4,3,4,4),NFLL
    3   LEFT=0.
    4   GO TO (6,5,5,6,5,5,6),NFLT
    5   TOP=0.
    6   GO TO (8,7,7,8,7,8,7),NFLB
    7   BOT=0.
8     CONTINUE
        CONV(IJ)=DELG*RGP(IJ)
      RBETA=EP(IJ)*RAGS+0.5*(DTODZ*DTODZ*((EP(IJ)+EP(IJT))*TOP+
     &(EP(IJ)+EP(IJB))*BOT)+DTORDR(I)*DTODR*(RB(I)*(EP(IJ)+EP(IJR))*
     &RIGHT+RB(I-1)*(EP(IJ)+EP(IJL))*LEFT))
        ABETA(IJ)=1./RBETA
 10   CONTINUE
      RETURN
      END
      SUBROUTINE CNVERT
      INCLUDE 'GCOME.FOR'
      DIMENSION CI(4)
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C Calculate initial values of the microscopic & macroscopic densities
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      CI(1)=P(IJ)
      CI(2)=TEMPO
      CI(3)=EP(IJ)
      CI(4)=TEMPO
      ROG(IJ)=CI(1)/(C(1)*CI(2))
      RGP(IJ)=ROG(IJ)*CI(3)
      RGPN(IJ)=RGP(IJ)
      DO 10 K=1,NSOLID
      RLKN(K,IJ)=RLK(K,IJ)
10    CONTINUE
      RETURN
      END
      SUBROUTINE EOSG(NR,NT,NC)
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the gas density and the speed of sound in gas
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      IF(NR.GT.0) ROG(IJ)=P(IJ)/(C(1)*TG)
      IF(NC.GT.1) RAGS=ROG(IJ)/P(IJ)
```

```
      RETURN
      END
      SUBROUTINE FLIC
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C/ Set cell flags based upon input data
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      COMMON/KFLIO/KFLIN(4),KFLOUT(4)
      RAD=DR*IB
      ZAX=DZ*JB
      J1=1
      J2=5
      J3=9
      J4=13
      YTS=-.5*DZ
      DO 150 J=1,JB2
      XTS=-.5*DR
      YTE=YTS+DZ*(J-1)
      DO 150 I=1,IB2
        XTE=XTS+DR*(I-1)
        IJ=I+(J-1)*IB2
C
C     Initialize all cell flags to '1'. This value will be changed
C     for non-fluid cells
C
        FL(IJ)=1
        IF (XTE.LT.0.) GO TO 10
        IF (XTE.GT.RAD) GO TO 30
        IF (YTE.GT.ZAX) GO TO 70
        IF (YTE.LT.0.) GO TO 110
        GO TO 150
C
C     Sets flag for left column( I = 1)
C
   10   IF (YTE.LT.FLO(J2)) GO TO 12
        IF (YTE.LT.FLO(J2+1)) GO TO 16
        J2=J2+2
        IF (J2.GT.7) J2=7
        IF (YTE.GT.FLO(J2).AND.YTE.LT.FLO(J2+1)) GO TO 16
   12   IF (NSL(2).EQ.1) GO TO 14
        IJ=I+(J-1)*IB2
        FL(IJ)=2
        GO TO 150
  14    IJ=I+(J-1)*IB2
        FL(IJ)=3
        GO TO 150
  16    IJ=I+(J-1)*IB2
        IF(J2.EQ.5)FL(IJ)=KFLIN(3)
        IF(J2.EQ.7)FL(IJ)=KFLIN(4)
        GO TO 150
C
C     Set flags for right column( I = IB2)
C
```

```
   30   IF (YTE.LT.FLO(J4)) GO TO 40
        IF (YTE.LT.FLO(J4+1)) GO TO 60
        J4=J4+2
        IF(J4.GT.15) J4=15
        IF (YTE.GT.FLO(J4).AND.YTE.LT.FLO(J4+1)) GO TO 60
   40   IF (NSL(4).EQ.1) GO TO 50
        IJ=I+(J-1)*IB2
        FL(IJ)=2
        GO TO 150
   50   IJ=I+(J-1)*IB2
        FL(IJ)=3
        GO TO 150
   60   IJ=I+(J-1)*IB2
        IF(J4.EQ.13)FL(IJ)=KFLOUT(3)
        IF(J4.EQ.15)FL(IJ)=KFLOUT(4)
        GO TO 150
C
C     Set flags for top ( J = JB2)
C
   70   IF (XTE.LT.FLO(J3)) GO TO 80
        IF (XTE.LT.FLO(J3+1)) GO TO 100
        J3=J3+2
        IF(J3.GT.11) J3=11
        IF(XTE.GT.FLO(J3).AND.XTE.LT.FLO(J3+1)) GO TO 100
   80   IF (NSL(3).EQ.1) GO TO 90
        IJ=I+(J-1)*IB2
        FL(IJ)=2
        GO TO 150
  90    IJ=I+(J-1)*IB2
        FL(IJ)=3
        GO TO 150
  100   IJ=I+(J-1)*IB2
        IF(J3.EQ.9)FL(IJ)=KFLOUT(1)
        IF(J3.EQ.11)FL(IJ)=KFLOUT(2)
        GO TO 150
C
C     Set flags for bottom ( J = 1) row
C
  110   IF(XTE.LT.FLO(J1)) GO TO 120
        IF(XTE.LT.FLO(J1+1)) GO TO 140
        J1=J1+2
        IF (J1.GT.3) J1=3
        IF(XTE.GT.FLO(J1).AND.XTE.LT.FLO(J1+1)) GO TO 140
  120   IF (NSL(1).EQ.1) GO TO 130
        IJ=I+(J-1)*IB2
        FL(IJ)=2
        GO TO 150
  130   IJ=I+(J-1)*IB2
        FL(IJ)=3
        GO TO 150
  140   IJ=I+(J-1)*IB2
        IF(J1.EQ.1)FL(IJ)=KFLIN(1)
        IF(J1.EQ.3)FL(IJ)=KFLIN(2)
```

```
  150 CONTINUE
      IB1JB2=IB1+(JB2-1)*IB2
      IB2JB1=IB2+(JB1-1)*IB2
      IB2JB2=IB2+(JB2-1)*IB2
      IF((FL(IB1JB2).EQ.4).AND.(FL(IB2JB1).EQ.4)) FL(IB2JB2)=4
      IF(FL(IB1JB2).EQ.7.AND.FL(IB2JB1).EQ.7) FL(IB2JB2)=7
      IF (NO.LE.0) RETURN
C
C     Set flags for obstacle cells
C
      DO 300 K=1,NO
        X1=OB(1,K)
        X2=OB(2,K)
        Y1=OB(3,K)
        Y2=OB(4,K)
        YTS=.5*DZ
        DO 290 J=2,JB1
        XTS=.5*DR
        YTE=YTS+DZ*(FLOAT(J-2))
        DO 290 I=2,IB1
          XTE=XTS+DR*(FLOAT(I-2))
          IF (XTE.LT.X1) GO TO 290
          IF (XTE.GT.X2) GO TO 290
          IF (YTE.LT.Y1) GO TO 290
          IF (YTE.GT.Y2) GO TO 290
          IJ=I+(J-1)*IB2
          FL(IJ)=2
          IF(K.GT.1.AND.FL(IJ-1).EQ.1)FL(IJ)=5
          IF(NSO(K).EQ.1) GO TO 280
          GO TO 290
  280     FL(IJ)=3
          IF(K.GT.1.AND.FL(IJ-1).EQ.1)FL(IJ)=5
  290   CONTINUE
  300 CONTINUE
      RETURN
      END
      SUBROUTINE INDEX
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate indices for array quantities
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      IMJP=IJ-1+IB2
      IJTL=IMJP
      IF((FL(IMJP).EQ.2).OR.(FL(IMJP).EQ.3)) IJTL=IJ+IB2
      IPJM=IJ+1-IB2
      IJBR=IPJM
      IF((FL(IPJM).EQ.2).OR.(FL(IPJM).EQ.3)) IJBR=IJ+1
      IJRR=IJ+2
      IF(I.EQ.IB1) IJRR=IJ+1
      IF((FL(IJRR).EQ.2).OR.(FL(IJRR).EQ.3)) IJRR=IJ+1
      IJTT=IJ+IB2+IB2
      IF(J.EQ.JB1) IJTT=IJ+IB2
```

```
      IF((FL(IJTT).EQ.2).OR.(FL(IJTT).EQ.3)) IJTT=IJ+IB2
      ENTRY INDEXA
      IPJ=IJ+1
      IJP=IJ+IB2
      IMJ=IJ-1
      IJM=IJ-IB2
      IMJM=IJM-1
      IPJP=IJP+1
      IJR=INDS(IJ,1)
      IJL=INDS(IJ,2)
      IJT=INDS(IJ,3)
      IJB=INDS(IJ,4)
      IJTR=INDS(IJ,5)
      RETURN
      END
      SUBROUTINE INTER
C
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate particle-particle friction coefficient
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      KS=1
      APPU(KS,IJ)=0.0
      APPV(KS,IJ)=0.0
      DO 100 K=2,NPHASE
C     CALCULATE PARTICLE TO PARTICLE INTERACTION********
      DO 98 KK=1,K
      KS = KK + K*(K-1)/2
      IF(KK.EQ.1)THEN
      APPU(KS,IJ)=-KPGU(K-1,IJ)*DT
      APPV(KS,IJ)=-KPGV(K-1,IJ)*DT
      ELSEIF(KK.EQ.K)THEN
      APPU(KS,IJ)=0.0
      APPV(KS,IJ)=0.0
      ELSE
      DV=ABS(VK(K-1,IJ)-VK(KK-1,IJ)+VK(K-1,IJM)-VK(KK-1,IJM))*0.5
      DU=ABS(UK(K-1,IJ)-UK(KK-1,IJ)+UK(K-1,IMJ)-UK(KK-1,IMJ))*0.5
      IF(DK(K-1).GT.DK(KK-1))THEN
      K1=K-1
      K2=KK-1
      ELSE
      K1=KK-1
      K2=K-1
      ENDIF
      EP1=RLK(K1,IJ)/RL(K1)
      EP2=RLK(K2,IJ)/RL(K2)
      EPSUM = EP1+EP2
      IF(EPSUM.NE.0.0)THEN
      XBAR=EP1/EPSUM
      IF(XBAR.LE.PHILIM(K-1,KK-1))THEN
      EPKL = EPSL(K-1,KK-1)*XBAR/PHI(K1) +PHI(K2)
      ELSE
      EPKL = EPSU(K-1,KK-1)*(1.0-XBAR) + PHI(K1)
```

```
      ENDIF
      EPKL = EPKL**(1.0/3.0)
      EPSUM  = EPSUM**(1.0/3.0)
      EPDIF = EPKL-EPSUM
      IF(EPDIF.LE.0.0)EPDIF=1.0D-30
      CON = EP1*EP2*DKF(K-1,KK-1)*(EPSUM+3.0*EPKL)/EPDIF
      ELSE
      CON = 0.0
      ENDIF
      APPU(KS,IJ)=-CON*DU
      APPV(KS,IJ)=-CON*DV
      ENDIF
98    CONTINUE
100   CONTINUE
      DO 105 K=1,NPHASE
      SUMX=0.0
      SUMY=0.0
      DO 101 KK=1,K
      KS = KK + K*(K-1)/2
      SUMX=SUMX+APPU(KS,IJ)
      SUMY=SUMY+APPV(KS,IJ)
101   CONTINUE
      DO 102 KK=K,NPHASE
      KS = K + KK*(KK-1)/2
      SUMX=SUMX+APPU(KS,IJ)
      SUMY=SUMY+APPV(KS,IJ)
102   CONTINUE
      KS=K*(K+1)/2
      APPU(KS,IJ)=-SUMX
      APPV(KS,IJ)=-SUMY
105   CONTINUE
      RETURN
      END
      SUBROUTINE ITER
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Perform the iterative solution of the difference equations
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      PARAMETER (INMAX = 5,OUTMAX = 20, OMEGA = 1.05)
      NIT=0
      MUSTIT=1
      RRT=1./(C(1)*TG)
    1 NIT=NIT+1
      DO 100 J=2,JB1
      DO 100 I=2,IB1
        IJ=I+(J-1)*IB2
        IF(FL(IJ).NE.1) GO TO 100
        LOOP=0
        KLOOP=0
        KROS=-1
        CALL INDEXA
        DG=RGP(IJ)-RGPN(IJ)+DTORDR(I)*(RGFR(IJ)-RGFR(IMJ))+DTODZ*
     1  (RGFT(IJ)-RGFT(IJM))
```

```
      TARGET = 0.0
        DGORIG=DG
        IF(ABS(DG).LE.CONV(IJ)) GO TO 78
        MUSTIT=1
        D3=DG
        P3=P(IJ)
        IF(NIT.EQ.1)GOTO55
   10   IF(D3.GT.0.0)GOTO11
        D2=D3
        P2=P3
        IF(KROS.EQ.-1)KROS=1
        IF(KROS.EQ.0)KROS=2
        GOTO12
   11   D1=D3
        P1=P3
        IF(KROS.EQ.-1)KROS=0
        IF(KROS.EQ.1)KROS=2
   12   IF(KROS.EQ.3)GOTO54
        IF(KROS.EQ.2)GOTO13
        DP=-D3*OMEGA*ABETA(IJ)
        IF(-DP*DSIGN(1.D0,D3).GT.0.25*P3) DP=-0.25*DSIGN(1.D0,
     1  D3)*P3
        P(IJ)=P(IJ) + DP
C          IF(P(IJ).LT.0.0) WRITE(1,93) IJ,P(IJ),DP,ABETA(IJ),D3
C 93       FORMAT(2X,'&&&&&&&&',2X,I3,4(2X,1PE12.4))
        GOTO54
   13   P(IJ)=(D1*P2-D2*P1)/(D1-D2)
        ABETA(IJ)=(P1-P2)/(D1-D2)
C          IF(P(IJ).LE.0.0)WRITE(1,92)IJ,P1,P2,D1,D2,ABETA(IJ)
C 92       FORMAT(2X,'&&&&&&&',I4,5(2X,1PE12.4))
        KROS=3
   54   P3=P(IJ)
 55     CONTINUE
        ROG(IJ)=P(IJ)*RRT
        RGP(IJ)=EP(IJ)*ROG(IJ)
      CALL MATS
C     WRITE(6,200)J,NIT,LOOP
C     WRITE(6,110)(AV(I1,1),AV(I1,2),BV(I1),I1=1,2)
C     WRITE(6,110)(AV1(I1,1),AV1(I1,2),BV1(I1),I1=1,2)
C110  FORMAT(3(4X,G14.7))
C200  FORMAT(/,4X,3(I2,3X))
      CALL VELSK
C     WRITE(6,110)VG(IJ),VK(1,IJ),P(IJ)
      CALL MASFK
78    RLX=0.0
      DO 79 K=1,NSOLID
      RLK(K,IJ)=RLKN(K,IJ)-DTORDR(I)*(RLFRK(K,IJ)-RLFRK(K,IMJ))
     &          -DTODZ*(RLFTK(K,IJ)-RLFTK(K,IJM))
      RLX=RLX+RLK(K,IJ)/RL(K)
79    CONTINUE
      EPX=RLX
      IF(EPX.LT.0.0)THEN
      EPX=0.0
```

```
      DO 80 K=1,NSOLID
      RLK(K,IJ)=0.0
80    CONTINUE
      ELSEIF(EPX.GT.0.8)THEN
      DIV=0.8/EPX
      EPX=0.8
      DO 81 K=1,NSOLID
      RLK(K,IJ)=RLK(K,IJ)*DIV
81    CONTINUE
      ENDIF
      EP(IJ)=1.-EPX
        RGP(IJ)=ROG(IJ)*EP(IJ)
        IF(ABS(DG).LE.CONV(IJ))GOTO100
        CALL MASFG
        DG=RGP(IJ)-RGPN(IJ)+DTORDR(I)*(RGFR(IJ)-RGFR(IMJ))+DTODZ*
     1  (RGFT(IJ)-RGFT(IJM))
        IF((ABS(DG).LE.CONV(IJ)).AND.(ABS(DG).LT.ABS(DGORIG)))
     1     GO TO 100
        IF((NIT.EQ.1).AND.(LOOP.EQ.0)) DGORIG=DG
        D3=DG
        LOOP=LOOP+1
        IF((KROS.LT.2).AND.(LOOP.EQ.INMAX)) ABETA(IJ)=.5*INMAX
       &*ABETA(IJ)
        IF(LOOP.EQ.INMAX) GO TO 100
        IF(KROS.EQ.3)CALL NEWP
        GOTO10
  100 CONTINUE
      IF(MUSTIT.EQ.0) RETURN
      MUSTIT=0
      IF(NIT.LT.OUTMAX)GOTO 1
      WRITE(6,'(A,G11.4)')' MAX. NO OF ITERTNS EXCEEDED AT TIME =',
     &TIME
      RETURN
      END
      SUBROUTINE KDRAGG(DRAG,DV,ISH)
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Gas-solids drag for low solids concentration
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DRAG=0.0
      DRACOE=4.4E-1
C     IF(ISH.EQ.1)THEN
C     EPX=(EP(IJ)+EP(IJT))*0.5
C     RGX=(ROG(IJ)+ROG(IJT))*0.5
C     RLKX=(RLK(K,IJ)+RLK(K,IJT))*0.5
C     ELSE
C     EPX=(EP(IJ)+EP(IJR))*0.5
C     RGX=(ROG(IJ)+ROG(IJR))*0.5
C     RLKX=(RLK(K,IJ)+RLK(K,IJR))*0.5
C     ENDIF
      EPX=EP(IJ)
      RGX=ROG(IJ)
      RLKX=RLK(K,IJ)
```

```
      REYNUM=EPX*RGX*DK(K)*PHIS*DV/(MMUG)
      IF(REYNUM.GE.0.0.AND.REYNUM.LT.1.0E-3)REYNUM=1.0E-3
      IF(REYNUM.GT.1000.0)GOTO 115
      DRACOE=(24.0/REYNUM)*(1.0+0.15*REYNUM**0.687)
 115  CONTINUE
      DRDVEP=DRACOE*DV/EPX**2.7
      IF(DRDVEP.GT.1.0E30)GO TO 135
      DRAG=0.75*DRDVEP*RGX
     1     /(DK(K)*PHIS)
      DRAG=RLKX*DRAG*EPX/RL(K)
      GO TO 125
 135  DRAG=1.0E30
 125  CONTINUE
      RETURN
      END
      SUBROUTINE KDRAGL(DRAG,DV,ISH)
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Gas-solids drag for high solids concentration
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     IF(ISH.EQ.1)THEN
C     EPX=(EP(IJ)+EP(IJT))*0.5
C     RGX=(ROG(IJ)+ROG(IJT))*0.5
C     RLKX=(RLK(K,IJ)+RLK(K,IJT))*0.5
C     ELSE
C     EPX=(EP(IJ)+EP(IJR))*0.5
C     RGX=(ROG(IJ)+ROG(IJR))*0.5
C     RLKX=(RLK(K,IJ)+RLK(K,IJR))*0.5
C     ENDIF
      EPX=EP(IJ)
      RGX=ROG(IJ)
      RLKX=RLK(K,IJ)
      DENOM2=EPX*DK(K)*PHIS
      DENOM1=DENOM2*DENOM2
      DRAG=150.0*(1.0-EPX)*MMUG/DENOM1
     1     +1.75*RGX*DV/DENOM2
      DRAG=DRAG*RLKX*EPX/RL(K)
      CONTINUE
      RETURN
      END
      SUBROUTINE KDRAGS
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Select KDRAGG or KDRAGL for Ggas-solids friction
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DO 130 K=1,NSOLID
      DVGS=ABS((VG(IJ)-VK(K,IJ)+VG(IJM)-VK(K,IJM))*0.5)
C     DVGS=ABS(VG(IJ)-VK(K,IJ))
      DUGS=ABS((UG(IJ)-UK(K,IJ)+UG(IMJ)-UK(K,IMJ))*0.5)
C     DUGS=ABS(UG(IJ)-UK(K,IJ))
      IF(EP(IJ).LT.0.8)GOTO 115
      ISH=1
      CALL KDRAGG(DRAGY,DVGS,ISH)
```

```
      ISH=2
      CALL KDRAGG(DRAGX,DUGS,ISH)
      GOTO 125
 115  CONTINUE
      ISH=1
      CALL KDRAGL(DRAGY,DVGS,ISH)
      ISH=2
      CALL KDRAGL(DRAGX,DUGS,ISH)
 125  CONTINUE
      KPGU(K,IJ)=DRAGX
      KPGV(K,IJ)=DRAGY
130   CONTINUE
      RETURN
      END
      SUBROUTINE MASFG
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculates mass fluxes for the gas
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      IF(UG(IMJ).GE.0.)THEN
      RGFR(IMJ)=UG(IMJ)*RGP(IJL)*RB(I-1)
      ELSE
       RGFR(IMJ)=UG(IMJ)*RGP(IJ)*RB(I-1)
      ENDIF
      IF(VG(IJM).GE.0.)THEN
       RGFT(IJM)=VG(IJM)*RGP(IJB)
      ELSE
       RGFT(IJM)=VG(IJM)*RGP(IJ)
      ENDIF
      ENTRY MASFGA
      IF(UG(IJ).GE.0.)THEN
       RGFR(IJ)=UG(IJ)*RGP(IJ)*RB(I)
      ELSE
       RGFR(IJ)=UG(IJ)*RGP(IJR)*RB(I)
      ENDIF
      IF(VG(IJ).GE.0.)THEN
       RGFT(IJ)=VG(IJ)*RGP(IJ)
      ELSE
       RGFT(IJ)=VG(IJ)*RGP(IJT)
      ENDIF
      RETURN
      END
      SUBROUTINE MASFK
C
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculates mass fluxes for the solids
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DO 10 K=1,NSOLID
      IF(UK(K,IMJ).GE.0.)THEN
      RLFRK(K,IMJ)=UK(K,IMJ)*RLK(K,IJL)*RB(I-1)
      ELSE
      RLFRK(K,IMJ)=UK(K,IMJ)*RLK(K,IJ)*RB(I-1)
```

```
      ENDIF
      IF(VK(K,IJM).GE.0.)THEN
      RLFTK(K,IJM)=VK(K,IJM)*RLK(K,IJB)
      ELSE
      RLFTK(K,IJM)=VK(K,IJM)*RLK(K,IJ)
      ENDIF
10    CONTINUE
C
      ENTRY MASFKA
      DO 20 K=1,NSOLID
      IF(UK(K,IJ).GE.0.)THEN
      RLFRK(K,IJ)=UK(K,IJ)*RLK(K,IJ)*RB(I)
      ELSE
      RLFRK(K,IJ)=UK(K,IJ)*RLK(K,IJR)*RB(I)
      ENDIF
      IF(VK(K,IJ).GE.0.)THEN
      RLFTK(K,IJ)=VK(K,IJ)*RLK(K,IJ)
      ELSE
      RLFTK(K,IJ)=VK(K,IJ)*RLK(K,IJT)
      ENDIF
20    CONTINUE
      RETURN
      END
      SUBROUTINE MATS
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculates the matrix components for velocity calculation
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      GL =0.0
      GB =0.0
      GJ =0.0
      IF(EP(IJL).LT.EPSTAR)GL=1.5D-3*EXP(500.*(EPSTAR-EP(IJL)))
      IF(EP(IJB).LT.EPSTAR)GB=1.5D-3*EXP(500.*(EPSTAR-EP(IJB)))
      IF(EP(IJ).LT.EPSTAR)GJ=1.5D-3*EXP(500.*(EPSTAR-EP(IJ)))
      GU=0.5*(GL+GJ)
      GV=0.5*(GB+GJ)
      DTPODR=DTODR*(P(IJ)-P(IJL))
      DTPODZ=DTODZ*(P(IJ)-P(IJB))
      DTEODR=DTODR*(EP(IJ)-EP(IJL))*GU
      DTEODZ=DTODZ*(EP(IJ)-EP(IJB))*GV
      BU1(1)=RUG(IMJ)-DTPODR*(EP(IJL)+EP(IJ))*0.5
      BV1(1)=RVG(IJM)-DTPODZ*(EP(IJB)+EP(IJ))*0.5
      AU1(1,1)=(APPU(1,IJL)+APPU(1,IJ)+RGP(IJL)+RGP(IJ))*0.5
      AV1(1,1)=(APPV(1,IJB)+APPV(1,IJ)+RGP(IJB)+RGP(IJ))*0.5
      DO 130 K=2,NPHASE
      BU1(K)=RUK(K-1,IMJ)+(DTEODR-DTPODR)*(RLK(K-1,IJL)+RLK(K-1,IJ))
     &          *0.5/RL(K-1)
      BV1(K)=RVK(K-1,IJM)+(DTEODZ-DTPODZ)*(RLK(K-1,IJB)+RLK(K-1,IJ))
     &          *0.5/RL(K-1)
      DO 110 KK=1,K
      KS = KK + K*(K-1)/2
      AU1(K,KK)=(APPU(KS,IJL)+APPU(KS.IJ))*0.5
C     AU1(K,KK)=APPU(KS,IJL)
```

```
      AU1(KK,K)=AU1(K,KK)
      AV1(K,KK)=(APPV(KS,IJB)+APPV(KS,IJ))*0.5
C     AV1(K,KK)=APPV(KS,IJB)
      AV1(KK,K)=AV1(K,KK)
110   CONTINUE
      AU1(K,K)=AU1(K,K)+(RLK(K-1,IJL)+RLK(K-1,IJ))*0.5
      AV1(K,K)=AV1(K,K)+(RLK(K-1,IJB)+RLK(K-1,IJ))*0.5
130   CONTINUE
      ENTRY MATSA
      GL =0.0
      GB =0.0
      GJ =0.0
      IF(EP(IJR).LT.EPSTAR)GL=1.5D-3*EXP(500.*(EPSTAR-EP(IJR)))
      IF(EP(IJT).LT.EPSTAR)GB=1.5D-3*EXP(500.*(EPSTAR-EP(IJT)))
      IF(EP(IJ).LT.EPSTAR)GJ=1.5D-3*EXP(500.*(EPSTAR-EP(IJ)))
      GU=0.5*(GL+GJ)
      GV=0.5*(GB+GJ)
      DTPODR=DTODR*(P(IJR)-P(IJ))
      DTPODZ=DTODZ*(P(IJT)-P(IJ))
      DTEODR=DTODR*(EP(IJR)-EP(IJ))*GU
      DTEODZ=DTODZ*(EP(IJT)-EP(IJ))*GV
      BU(1)=RUG(IJ)-DTPODR*(EP(IJ)+EP(IJR))*0.5
      BV(1)=RVG(IJ)-DTPODZ*(EP(IJ)+EP(IJT))*0.5
      AU(1,1)=(APPU(1,IJ)+APPU(1,IJR)+RGP(IJ)+RGP(IJR))*0.5
      AV(1,1)=(APPV(1,IJ)+APPV(1,IJT)+RGP(IJ)+RGP(IJT))*0.5
      DO 230 K=2,NPHASE
      BU(K)=RUK(K-1,IJ)+(DTEODR-DTPODR)*(RLK(K-1,IJ)+RLK(K-1,IJR))
     &            *0.5/RL(K-1)
      BV(K)=RVK(K-1,IJ)+(DTEODZ-DTPODZ)*(RLK(K-1,IJ)+RLK(K-1,IJT))
     &           *0.5/RL(K-1)
      DO 210 KK=1,K
      KS = KK + K*(K-1)/2
      AU(K,KK)=(APPU(KS,IJ)+APPU(KS,IJR))*0.5
C     AU(K,KK)=APPU(KS,IJ)
      AU(KK,K)=AU(K,KK)
      AV(K,KK)=(APPV(KS,IJ)+APPV(KS,IJT))*0.5
C     AV(K,KK)=APPV(KS,IJ)
      AV(KK,K)=AV(K,KK)
210   CONTINUE
      AU(K,K)=AU(K,K)+(RLK(K-1,IJ)+RLK(K-1,IJR))*0.5
      AV(K,K)=AV(K,K)+(RLK(K-1,IJ)+RLK(K-1,IJT))*0.5
230   CONTINUE
C
      RETURN
      END
      SUBROUTINE NEWP
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C    Calculates the new estimate of advanced time pressure from
C    three (P,D) points.
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      IF(D1.NE.D3)PA=(D1*P3-D3*P1+TARGET*(P1-P3))/(D1-D3)
      IF((D1-TARGET)*(D3-TARGET).LE.0.) GO TO 1
```

```
      IF(D1.EQ.D3)PA=0.5*(P2+P3)
      IF(PA.LT.P2.OR.PA.GT.P3)PA=0.5*(P2+P3)
      PB=(D2*P3-D3*P2+TARGET*(P2-P3))/(D2-D3)
      GOTO10
    1 IF(D2.NE.D3) PB=(D2*P3-D3*P2+TARGET*(P2-P3))/(D2-D3)
      IF(D2.EQ.D3)PB=0.5*(P1+P3)
      IF(PB.LT.P3.OR.PB.GT.P1)PB=0.5*(P1+P3)
   10 P(IJ)=0.5*(PA+PB)
      RETURN
      END
      SUBROUTINE PROG
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Control the program flow and output
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      PARAMETER (FAC = 0.5)
      DIMENSION EPL(50)
C
      IJC=0
      IRST=0
      TDUMP1=TIME
      TPRI=TIME
      CYCL1=CYCLE
C     INITIALIZE  'GEP'
      DO 20 I=1,100
      XIJ=I
      EPX=XIJ/100.
      GEP(I)=0.0
C     GEP(I)=(10.**(-10.46*EPX+6.577))*10.       SEE MATS
20    CONTINUE
C
C     INITIALIZE 'PHI' THE MAX SINGLE PARTICLE PACKING
      DO 50 K=1,NSOLID
      PHI(K) = 0.6
50    CONTINUE
C
      DO 51 K=1,NSOLID
      DO 51 KK=1,NSOLID
      IF(DK(K).GT.DK(KK))THEN
      K1=K
      K2=KK
      ELSE
      K1=KK
      K2=K
      ENDIF
      DRATX=SQRT(DK(K2)/DK(K1))
      PHILIM(K,KK)=PHI(K1)/(PHI(K1)+(1.0-PHI(K1))*PHI(K2))
      EPSL(K,KK)=(PHI(K1)-PHI(K2)+(1.0-DRATX)*(1.0-PHI(K1))*PHI(K2))
     & *(PHI(K1)+(1.0-PHI(K1))*PHI(K2))
      EPSU(K,KK)=(1.0-DRATX)*(PHI(K1)+(1.0-PHI(K1))*PHI(K2))
51    CONTINUE
C     INITIALIZE 'INDS'
      DO 30 I=2,IB1
```

```
      DO 30 J=2,JB1
      IJ=I+(J-1)*IB2
      IPJ=IJ+1
      IJR=IPJ
      IJP=IJ+IB2
      IJT=IJP
      IMJ=IJ-1
      IJL=IMJ
      IJM=IJ-IB2
      IMJM=IJM-1
      IJB=IJM
      IPJP=IJP+1
      IJTR=IPJP
      IF((FL(IPJ).EQ.2).OR.(FL(IPJ).EQ.3)) IJR=IJ
      IF((FL(IMJ).EQ.2).OR.(FL(IMJ).EQ.3)) IJL=IJ
      IF((FL(IJP).EQ.2).OR.(FL(IJP).EQ.3)) IJT=IJ
      IF((FL(IJM).EQ.2).OR.(FL(IJM).EQ.3)) IJB=IJ
      NFLTR=FL(IPJP)
      IF(I.EQ.IB1.AND.J.EQ.JB1.AND.NFLTR.EQ.4)IJTR=IJ
      GO TO (29,21,21,29,29,29,29),NFLTR
   21 NFLT=FL(IJP)
      NFLR=FL(IPJ)
      GO TO (22,24,24,22,22,22,22),NFLT
   22 GO TO (25,26,26,25,25,25,25),NFLR
   24 GO TO (28,25,25,28,28,28,28),NFLR
   25 IJTR=IJ
      GO TO 29
   26 IJTR=IJP
      GO TO 29
   28 IJTR=IPJ
29    CONTINUE
      INDS(IJ,1)=IJR
      INDS(IJ,2)=IJL
      INDS(IJ,3)=IJT
      INDS(IJ,4)=IJB
      INDS(IJ,5)=IJTR
30    CONTINUE
      DO 40 K=1,NSOLID
      DO 40 KK=1,K
      DKF(K,KK)=FAC*DT*RL(K)*RL(KK)*(DK(K)+DK(KK))**2
     & /(RL(K)*DK(K)**3+RL(KK)*DK(KK)**3)
      DKF(KK,K)=DKF(K,KK)
40    CONTINUE
    1 CONTINUE
      DO 10 J=2,JB1
      DO 10 I=2,IB1
        IJ=I+(J-1)*IB2
        IF(FL(IJ).NE.1) GO TO 10
        CALL INDEXA
        RGPN(IJ)=RGP(IJ)
        EPN(IJ)=EP(IJ)
      DO 9 K=1,NSOLID
      RLKN(K,IJ)=RLK(K,IJ)
```

```
9     CONTINUE
        CALL EOSG(1,1,2)
   10 CONTINUE
      CALL BDRY
      IF(TIME+0.1*DT.LT.TPRI)GOTO 1350
      WRITE(6,547)TIME
      TPRI=TPRI+TPR
      WRITE(6,548)
      DO 325 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(P(IJL),IJL=IJ1,IJ2)
325   CONTINUE
      DO 328 K=1,NSOLID
      WRITE(6,549)K
      DO 328 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      IJD=IJ2-IJ1+1
      IJL1=0
      DO 327 IJL=IJ1,IJ2
      IJL1=IJL1+1
      EPL(IJL1)=RLK(K,IJL)/RL(K)
327   CONTINUE
      WRITE(6,550)(EPL(IJL),IJL=1,IJD)
328   CONTINUE
      WRITE(6,551)
      IJC=IJC+1
      DO 331 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(EP(IJL),IJL=IJ1,IJ2)
331   CONTINUE
      WRITE(6,552)
      DO 332 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(VG(IJL),IJL=IJ1,IJ2)
332   CONTINUE
      WRITE(6,553)
      DO 333 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(UG(IJL),IJL=IJ1,IJ2)
333   CONTINUE
      DO 336 K=1,NSOLID
      WRITE(6,556)K
      DO 336 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(VK(K,IJL),IJL=IJ1,IJ2)
336   CONTINUE
      DO 337 K=1,NSOLID
```

```
      WRITE(6,557)K
      DO 337 J=1,JB2
      IJ1=1+(JB2-J)*IB2
      IJ2=IB2+(JB2-J)*IB2
      WRITE(6,550)(UK(K,IJL),IJL=IJ1,IJ2)
337   CONTINUE
 1350 CONTINUE
      IB2JB2=IB2*JB2
      IF(TIME+0.1*DT.GT.TSTOP)GOTO 390
      IF(TIME+0.1*DT.GT.TDUMP1)GOTO 390
      GOTO 400
390   CONTINUE
      CALL TAPEWR
      IF(IRST .EQ. 2) REWIND(9)
      IRST=IRST+1
      IF(IRST.GE.3)IRST=0
      TDUMP1=TDUMP1+TDUMP
  400 CONTINUE
      IF(TIME+.1*DT.GE.TSTOP) GO TO 462
      CALL TILDE
      CALL BETAS
      CALL ITER
      TIME=TIME+DT
      CYCLE=CYCLE+1
      GOTO 1
462   CONTINUE
      RETURN
547   FORMAT(1X,///,1X,'@@@ TIME = ',G10.3)
548   FORMAT(1X,//,1X,'P',/)
549   FORMAT(1X,//,1X,'EP',I1,/)
550   FORMAT(1X,10(1X,G11.4,1X))
551   FORMAT(1X,//,1X,'EPSG',/)
552   FORMAT(1X,//,1X,'VG',/)
553   FORMAT(1X,//,1X,'UG',/)
556   FORMAT(1X,//,1X,'V',I1,/)
557   FORMAT(1X,//,1X,'U',I1,/)
      END
      SUBROUTINE SETC
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Set the C array used to store necessary constants
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     GAS CONSTANT FOR AIR
      C(1)=2.87E6
      MMUG=1.8E-4
      PHIS=1.00E00
      RLIM=0.00001
C     INITIALIZE 'EPSTAR'
      EPSTAR =0.4
      C(1)=C(1)*SCALE(4)/SCALE(2)**2
      RETURN
      END
      SUBROUTINE SETUP (JH)
```

```
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Initialize the radii and all the field variables
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DIMENSION RLX(20)
C
      CALL SETC
      RDR=1./DR
      RDZ=1./DZ
      DTODZ=DT*RDZ
      DTODR=DT*RDR
      RDR2=RDR*RDR
      RDZ2=RDZ*RDZ
      IF(ITC.EQ.0) GO TO 15
      EX=FLOAT(ITC)
      R(1)=-(0.5/EX)*DR**EX
      RB(1)=0.
      RRB(1)=0.
      DO 10 I=2,IB2
        R(I)=(((FLOAT(I))-1.5)*DR)**EX
        RB(I)=(((FLOAT(I))-1.0)*DR)**EX
        RRB(I)=1./RB(I)
        RRIDR(I)=RDR/R(I)
        DTORDR(I)=DT*RRIDR(I)
        DTOBDR(I)=DTODR*RRB(I)
   10 CONTINUE
      GO TO 25
   15 DO 20 I=1,IB2
        R(I)=1.
        RB(I)=1.
        RRB(I)=1.
        RRIDR(I)=RDR
        DTORDR(I)=DT*RDR
        DTOBDR(I)=DTORDR(I)
   20 CONTINUE
   25 CONTINUE
      WRITE(6,660)
      I1=JB2*IB2+1
      DO 30 J=1,JB2
        KPR=JB2-J+1
        IKPR=IB2+(KPR-1)*IB2
        I1=I1-IB2
        I2=I1+IB2-1
        WRITE(6,650)(FL(IKPR),IKPR=I1,I2)
30    CONTINUE
      WRITE(6,660)
      DPT=DZ*PO*(-GRAV)/(C(1)*TG)
      WGY=VO*PO/(C(1)*TG)
      JH=JH+1
      POM=PO-DPT
      DO 77 J=JB2,2,-1
      EPX=EPO
      IF(J.GT.JH)EPX=1.0
```

```
      POM=POM+DPT*EPX
C>>>
```

```
        TG=TEMINL
        TL=TEMINL
        EP(IJ)=EPINL
        EPN(IJ)=EP(IJ)
        UG(IJ)=UINL
        VG(IJ)=VINL
      DO 52 K=1,NSOLID
      RLK(K,IJ)=RL(K)*(1.0-EPINL)
      VK(K,IJ)=VINL
      UK(K,IJ)=0.0
52    CONTINUE
        GOTO57
 55   P(IJ)=PINR
        TG=TEMINR
        TL=TEMINR
        EP(IJ)=EPINR
        EPN(IJ)=EP(IJ)
        UG(IJ)=UINR
        VG(IJ)=VINR
      DO 56 K=1,NSOLID
      RLK(K,IJ)=0.0
      VK(K,IJ)=0.0
      UK(K,IJ)=0.0
56    CONTINUE
  57    CALL CNVERT
        IJR=IJ+1
        IJT=IJ+IB2
        CALL MASFGA
      CALL MASFKA
        CALL STXTRA
   60 CONTINUE
      DO 70 J=2,JB1
        I=1
        IJ=I+(J-1)*IB2
        IF(FL(IJ).NE.5.AND.FL(IJ).NE.6)GO TO 70
        DIST=(J-1)*DZ-0.1*DZ
        IF(DIST.GT.FLO(7).AND.DIST.LT.FLO(8))GOTO65
        P(IJ)=PINB
        TG=TEMINB
        TL=TEMINB
        EP(IJ)=EPINB
        EPN(IJ)=EP(IJ)
        UG(IJ)=UINB
        VG(IJ)=VINB
      DO 61 K=1,NSOLID
      RLK(K,IJ)=RL(K)*(1.0-EPINB)
      VK(K,IJ)=VINB
      UK(K,IJ)=UINB
61    CONTINUE
        GOTO 67
   65   P(IJ)=PINT
        TG=TEMINT
        TL=TEMINB
```

```
        EP(IJ)=EPINT
      DO 66 K=1,NSOLID
      VK(K,IJ)=0.0
      UK(K,IJ)=0.0
66    CONTINUE
        VG(IJ)=VINT
        EPN(IJ)=EP(IJ)
        UG(IJ)=UINT
   67   CALL  CNVERT
        IJR=IJ+1
        IJT=IJ+IB2
        CALL MASFGA
      CALL MASFKA
        CALL STXTRA
   70 CONTINUE
  650 FORMAT (1X,63I2)
  660 FORMAT (///)
      RETURN
      END
      SUBROUTINE STXTRA
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Initialize constant pressure outflow boundary cells
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      IJ=I+(J-1)*IB2
      IMJ=IJ-1
      IJM=IJ-IB2
      IF(I.EQ.1)IMJ=IJ
      IF(J.EQ.1)IJM=IJ
      EPN(IJ)=EP(IJ)
      CALL EOSG(1,1,1)
      CALL KDRAGS
      CALL INTER
      RETURN
      END
      SUBROUTINE TAPERD
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Read the restart file
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      READ(9)TIME,IB2,JB2,NSOLID
      IB2JB2=IB2*JB2
      READ(9) (P(IJ),EP(IJ),UG(IJ),VG(IJ),IJ=1,IB2JB2)
      READ(9)((RLK(K,IJ),UK(K,IJ),VK(K,IJ),
     &  K=1,NSOLID),IJ=1,IB2JB2)
      RETURN
      END
      SUBROUTINE TAPEWR
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Write the restart file
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
      WRITE(9)TIME,IB2,JB2,NSOLID
      IB2JB2=IB2*JB2
      WRITE(9) (P(IJ),EP(IJ),UG(IJ),VG(IJ),IJ=1,IB2JB2)
      WRITE(9)((RLK(K,IJ),UK(K,IJ),VK(K,IJ),
     &  K=1,NSOLID),IJ=1,IB2JB2)
      RETURN
      END
      SUBROUTINE TILDE
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the tilde quantities
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      DO 10 J=2,JB1
      DO 10 I=2,IB1
        IJ=I+(J-1)*IB2
        IF(FL(IJ).NE.1) GO TO 10
        CALL INDEX
        CALL UGMOMF
      IGKU=(EP(IJ)+EP(IJR))*50
      IGKV=(EP(IJ)+EP(IJT))*50
        RUG(IJ)=.5*(RGP(IJ)+RGP(IJR))*UG(IJ)-DTOBDR (I)*(UGFR-UGFL)-
     1  DTODZ*(UGFT-UGFB(I))
        UGFL=UGFR
        UGFB(I)=UGFT
        CALL VGMOMF
        RVG(IJ)=0.5*(RGP(IJ)+RGP(IJT))*(VG(IJ)+GRAV*DT)-DTORDR(I)*
     1  (VGFR- VGFL)-DTODZ*(VGFT-VGFB(I))
        VGFL=VGFR
        VGFB(I)=VGFT
      DO 9 K=1,NSOLID
        CALL ULMOMF
        RULK1=.5*(RLK(K,IJ)+RLK(K,IJR))*UK(K,IJ)-DTOBDR(I)*(ULFR
     1   -ULFL(K))-DTODZ*(ULFT-ULFB(K,I))-GEP(IGKU)*DTODR*(RLK(K,IJR)
     2    -RLK(K,IJ))/RL(K)
      RUK(K,IJ)=RULK1
        ULFL(K)=ULFR
        ULFB(K,I)=ULFT
        CALL VLMOMF
        RVLK1=0.5*(RLK(K,IJ)+RLK(K,IJT))*(VK(K,IJ)+DT*GRAV)
     1   -DTORDR(I)*(VLFR-VLFL(K))-DTODZ*(VLFT-VLFB(K,I))
     2   -GEP(IGKV)*DTODZ*(RLK(K,IJT)-RLK(K,IJ))/RL(K)
      RVK(K,IJ)=RVLK1
        VLFL(K)=VLFR
        VLFB(K,I)=VLFT
9     CONTINUE
        CALL KDRAGS
      CALL INTER
   10 CONTINUE
      DO 20 J=2,JB1
      DO 20 I=2,IB1
        IJ=I+(J-1)*IB2
        IF(FL(IJ).NE.1) GO TO 20
```

```
        CALL INDEXA
      CALL MATSA
        CALL VELSK2
        CALL MASFGA
      CALL MASFKA
   20 CONTINUE
      RETURN
      END
      SUBROUTINE UGMOMF
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the radial momentum flux of gas
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DIMENSION CS(4)
      CS(1)=0.5*(UG(IJ)+UG(IPJ))
      IF(CS(1).GE.0.)UGFR=0.5*(RGP(IJ)+RGP(IJR))*UG(IJ)*CS(1)*R(I+1)
      IF(CS(1).LT.0.)UGFR=0.5*(RGP(IJR)+RGP(IJRR))*UG(IPJ)*CS(1)*R(I+1)
      CS(2)=0.5*(VG(IJ)+VG(IPJ))
      IF(CS(2).GE.0.) UGFT=0.5*(RGP(IJ)+RGP(IJR))*UG(IJ)*CS(2)
      IF(CS(2).LT.0.) UGFT=0.5*(RGP(IJT)+RGP(IJTR))*UG(IJP)*CS(2)
      IF(FL(IMJ).NE.1) GO TO 1
      IF(FL(IJM).NE.1) GO TO 2
      GO TO 3
    1 CS(3)=0.5*(UG(IJ)+UG(IMJ))
      IF(CS(3).GE.0.) UGFL=0.5*(RGP(IJ)+RGP(IJL))*UG(IMJ)*CS(3)*R(I)
      IF(CS(3).LT.0.) UGFL=0.5*(RGP(IJ)+RGP(IJR))*UG(IJ)*CS(3)*R(I)
      IF(FL(IJM).NE.1) GO TO 2
      GO TO 3
    2 CS(4)=0.5*(VG(IJM)+VG(IPJM))
      IF(CS(4).GE.0.) UGFB(I)=0.5*(RGP(IJB)+RGP(IJBR))*UG(IJM)*CS(4)
      IF(CS(4).LT.0.) UGFB(I)=0.5*(RGP(IJ)+RGP(IJR))*UG(IJ)*CS(4)
3     CONTINUE
      RETURN
      END
      SUBROUTINE ULMOMF
      INCLUDE 'GCOME.FOR'
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the radial momentum flux of solids
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
      DIMENSION CS(4)
      CS(1)=0.5*(UK(K,IJ)+UK(K,IPJ))
      IF(CS(1).GE.0.) ULFR=0.5*(RLK(K,IJ)+RLK(K,IJR))*UK(K,IJ)*CS(1)
     &  *R(I+1)
      IF(CS(1).LT.0.)ULFR=0.5*(RLK(K,IJR)+RLK(K,IJRR))*UK(K,IPJ)*CS(1)
     &  *R(I+1)
      CS(2)=0.5*(VK(K,IJ)+VK(K,IPJ))
      IF(CS(2).GE.0.)ULFT=0.5*(RLK(K,IJ)+RLK(K,IJR))*UK(K,IJ)*CS(2)
      IF(CS(2).LT.0.)ULFT=0.5*(RLK(K,IJT)+RLK(K,IJTR))*UK(K,IJP)*CS(2)
      IF(FL(IMJ).NE.1) GO TO 1
      IF(FL(IJM).NE.1) GO TO 2
      GOTO3
    1 CS(3)=0.5*(UK(K,IJ)+UK(K,IMJ))
```

```
      IF(CS(3).GE.0.) ULFL(K)=0.5*(RLK(K,IJ)+RLK(K,IJL))*UK(K,IMJ)
     &    *CS(3)*R(I)
      IF(CS(3).LT.0.) ULFL(K)=0.5*(RLK(K,IJ)+RLK(K,IJR))*UK(K,IJ)
     &   *CS(3)*R(I)
      IF(FL(IJM).NE.1)GOTO2
      GOTO3
    2 CS(4)=0.5*(VK(K,IJM)+VK(K,IPJM))
      IF(CS(4).GE.0.) ULFB(K,I)=0.5*(RLK(K,IJB)+RLK(K,IJBR))*UK(K,IJM)
     &    *CS(4)
      IF(CS(4).LT.0.) ULFB(K,I)=0.5*(RLK(K,IJ)+RLK(K,IJR))*UK(K,IJ)
     &       *CS(4)
3     CONTINUE
      RETURN
      END
      SUBROUTINE VELSK
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the velocities at the cell boundaries
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     USE GAUSS-JORDAN METHOD FOR MATRIX INVERSION
      FLL=FL(IMJ)
      IF(FLL.EQ.2)GOTO 200
      IF(FLL.EQ.3)GOTO 200
      IF(FLL.EQ.5)GOTO 200
      DO 136 K=2,NPHASE
      IF(AU1(K,K).GE.RLIM)GOTO 136
      DO 135 KK=1,NPHASE
       AU1(KK,KK)=AU1(KK,KK)+AU1(K,KK)
      AU1(K,KK)=0.0
      AU1(KK,K)=0.0
135   CONTINUE
      BU1(K)=0.0
136   CONTINUE
C
      DO 160 K=1,NPHASE
      IF(AU1(K,K).EQ.0.0)GOTO 160
      KP1=K+1
C     IF(KP1.GT.NPHASE)GOTO 141
      DIV=1./AU1(K,K)
      DO 140 KJ=KP1,NPHASE
C     AU1(K,KJ)=AU1(K,KJ)/AU1(K,K)
      AU1(K,KJ)=AU1(K,KJ)*DIV
140   CONTINUE
C141   BU1(K)=BU1(K)/AU1(K,K)
       BU1(K)=BU1(K)*DIV
C     AU1(K,K)=1.0
      AU1(K,K)=0.0
      DO 150 KI=1,NPHASE
C     IF(KI.EQ.K)GOTO 150
C     IF(AU1(KI,K).EQ.0.0)GOTO 150
C     IF(KP1.GT.NPHASE)GOTO 146
      AMUL=AU1(KI,K)
      DO 145 KJ=KP1,NPHASE
```

```
C       AU1(KI,KJ)=AU1(KI,KJ)-AU1(KI,K)*AU1(K,KJ)
        AU1(KI,KJ)=AU1(KI,KJ)-AMUL*AU1(K,KJ)
145     CONTINUE
C146     BU1(KI)=BU1(KI)-AU1(KI,K)*BU1(K)
         BU1(KI)=BU1(KI)-AMUL*BU1(K)
C       AU1(KI,K)=0.0
150     CONTINUE
160     CONTINUE
        UG(IMJ)=BU1(1)
        DO 165 K=2,NPHASE
        UK(K-1,IMJ)=BU1(K)
165     CONTINUE
200     CONTINUE
        FLB=FL(IJM)
        IF(FLB.EQ.2)GOTO 300
        IF(FLB.EQ.3)GOTO 300
        IF(FLB.EQ.5)GOTO 300
C
C
        DO 236 K=2,NPHASE
        IF(AV1(K,K).GE.RLIM)GOTO 236
        DO 235 KK=1,NPHASE
        AV1(KK,KK)=AV1(KK,KK)+AV1(K,KK)
        AV1(K,KK)=0.0
        AV1(KK,K)=0.0
235     CONTINUE
        BV1(K)=0.0
236     CONTINUE
C
        DO 260 K=1,NPHASE
        IF(AV1(K,K).EQ.0.0)GOTO 260
        KP1=K+1
C       IF(KP1.GT.NPHASE)GOTO 241
        DIV=1./AV1(K,K)
        DO 240 KJ=KP1,NPHASE
C       AV1(K,KJ)=AV1(K,KJ)/AV1(K,K)
        AV1(K,KJ)=AV1(K,KJ)*DIV
240     CONTINUE
C241     BV1(K)=BV1(K)/AV1(K,K)
         BV1(K)=BV1(K)*DIV
C       AV1(K,K)=1.0
        AV1(K,K)=0.0
        DO 250 KI=1,NPHASE
C       IF(KI.EQ.K)GOTO 250
C       IF(AV1(KI,K).EQ.0.0)GOTO 250
C       IF(KP1.GT.NPHASE)GOTO 246
        AMUL=AV1(KI,K)
        DO 245 KJ=KP1,NPHASE
C       AV1(KI,KJ)=AV1(KI,KJ)-AV1(KI,K)*AV1(K,KJ)
        AV1(KI,KJ)=AV1(KI,KJ)-AMUL*AV1(K,KJ)
245     CONTINUE
C246     BV1(KI)=BV1(KI)-AV1(KI,K)*BV1(K)
         BV1(KI)=BV1(KI)-AMUL*BV1(K)
```

```
C     AV1(KI,K)=0.0
250   CONTINUE
260   CONTINUE
      VG(IJM)=BV1(1)
      DO 265 K=2,NPHASE
      VK(K-1,IJM)=BV1(K)
265   CONTINUE
C
      ENTRY VELSK2
300   CONTINUE
      FLR=FL(IPJ)
      IF(FLR.EQ.2)GOTO 400
      IF(FLR.EQ.3)GOTO 400
      IF(FLR.EQ.5)GOTO 400
      DO 336 K=2,NPHASE
      IF(AU(K,K).GE.RLIM)GOTO 336
      DO 335 KK=1,NPHASE
      AU(KK,KK)=AU(KK,KK)+AU(K,KK)
      AU(K,KK)=0.0
      AU(KK,K)=0.0
335   CONTINUE
      BU(K)=0.0
336   CONTINUE
C
      DO 360 K=1,NPHASE
      IF(AU(K,K).EQ.0.0)GOTO 360
      KP1=K+1
C     IF(KP1.GT.NPHASE)GOTO 341
      DIV=1./AU(K,K)
      DO 340 KJ=KP1,NPHASE
C     AU(K,KJ)=AU(K,KJ)/AU(K,K)
      AU(K,KJ)=AU(K,KJ)*DIV
340   CONTINUE
C341   BU(K)=BU(K)/AU(K,K)
      BU(K)=BU(K)*DIV
C     AU(K,K)=1.0
      AU(K,K)=0.0
      DO 350 KI=1,NPHASE
C     IF(KI.EQ.K)GOTO 350
C     IF(AU(KI,K).EQ.0.0)GOTO 350
C     IF(KP1.GT.NPHASE)GOTO 346
      AMUL=AU(KI,K)
      DO 345 KJ=KP1,NPHASE
C     AU(KI,KJ)=AU(KI,KJ)-AU(KI,K)*AU(K,KJ)
      AU(KI,KJ)=AU(KI,KJ)-AMUL*AU(K,KJ)
345   CONTINUE
C346   BU(KI)=BU(KI)-AU(KI,K)*BU(K)
      BU(KI)=BU(KI)-AMUL*BU(K)
C     AU(KI,K)=0.0
350   CONTINUE
360   CONTINUE
      UG(IJ)=BU(1)
      DO 365 K=2,NPHASE
```

```
      UK(K-1,IJ)=BU(K)
365   CONTINUE
400   CONTINUE
      FLT=FL(IJP)
      IF(FLT.EQ.2)RETURN
      IF(FLT.EQ.3)RETURN
      IF(FLT.EQ.5)RETURN
C
C
      DO 436 K=2,NPHASE
      IF(AV(K,K).GE.RLIM)GOTO 436
      DO 435 KK=1,NPHASE
      AV(KK,KK)=AV(KK,KK)+AV(K,KK)
      AV(K,KK)=0.0
      AV(KK,K)=0.0
435   CONTINUE
      BV(K)=0.0
436   CONTINUE
C
      DO 460 K=1,NPHASE
      IF(AV(K,K).EQ.0.0)GOTO 460
      KP1=K+1
C     IF(KP1.GT.NPHASE)GOTO 441
      DIV=1./AV(K,K)
      DO 440 KJ=KP1,NPHASE
C     AV(K,KJ)=AV(K,KJ)/AV(K,K)
      AV(K,KJ)=AV(K,KJ)*DIV
440   CONTINUE
C441   BV(K)=BV(K)/AV(K,K)
      BV(K)=BV(K)*DIV
C     AV(K,K)=1.0
      AV(K,K)=0.0
      DO 450 KI=1,NPHASE
C     IF(KI.EQ.K)GOTO 450
C     IF(AV(KI,K).EQ.0.0)GOTO 450
C     IF(KP1.GT.NPHASE)GOTO 446
      AMUL=AV(KI,K)
      DO 445 KJ=KP1,NPHASE
C     AV(KI,KJ)=AV(KI,KJ)-AV(KI,K)*AV(K,KJ)
      AV(KI,KJ)=AV(KI,KJ)-AMUL*AV(K,KJ)
445   CONTINUE
C446   BV(KI)=BV(KI)-AV(KI,K)*BV(K)
      BV(KI)=BV(KI)-AMUL*BV(K)
C     AV(KI,K)=0.0
450   CONTINUE
460   CONTINUE
      VG(IJ)=BV(1)
      DO 465 K=2,NPHASE
      VK(K-1,IJ)=BV(K)
465   CONTINUE
      RETURN
      END
      SUBROUTINE VGMOMF
```

```
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the axial momentum flux of gas
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DIMENSION CS(4)
      CS(1)=0.5*(VG(IJ)+VG(IJP))
      IF(CS(1).GE.0.) VGFT=0.5*(RGP(IJ)+RGP(IJT))*VG(IJ)*CS(1)
      IF(CS(1).LT.0.) VGFT=0.5*(RGP(IJT)+RGP(IJTT))*VG(IJP)*CS(1)
      CS(2)=0.5*(UG(IJ)+UG(IJP))
      IF(CS(2).GE.0.)VGFR=0.5*(RGP(IJ)+RGP(IJT))*VG(IJ)*CS(2)*RB(I)
      IF(CS(2).LT.0.)VGFR=0.5*(RGP(IJR)+RGP(IJTR))*VG(IPJ)*CS(2)*RB(I)
      IF(FL(IMJ).NE.1) GO TO 1
      IF(FL(IJM).NE.1) GO TO 2
      GOTO3
    1 CS(3)=0.5*(UG(IMJ)+UG(IMJP))
      IF(CS(3).GE.0.)VGFL=0.5*(RGP(IJL)+RGP(IJTL))*VG(IMJ)*CS(3)*RB(I-1)
      IF(CS(3).LT.0.)VGFL=0.5*(RGP(IJ)+RGP(IJT))*VG(IJ)*CS(3)*RB(I-1)
      IF(FL(IJM).NE.1)GOTO2
      GOTO3
    2 CS(4)=0.5*(VG(IJM)+VG(IJ))
      IF(CS(4).GE.0.) VGFB(I)=0.5*(RGP(IJ)+RGP(IJB))*VG(IJM)*CS(4)
      IF(CS(4).LT.0.) VGFB(I)=0.5*(RGP(IJ)+RGP(IJT))*VG(IJ)*CS(4)
      RETURN
3     CONTINUE
      END
      SUBROUTINE VLMOMF
      INCLUDE 'GCOME.FOR'
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C     Calculate the axial momentum flux of solids
C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
      DIMENSION CS(4)
      CS(1)=0.5*(VK(K,IJ)+VK(K,IJP))
      IF(CS(1).GE.0.)VLFT=0.5*(RLK(K,IJ)+RLK(K,IJT))*VK(K,IJ)*CS(1)
      IF(CS(1).LT.0.)VLFT=0.5*(RLK(K,IJT)+RLK(K,IJTT))*VK(K,IJP)*CS(1)
      CS(2)=0.5*(UK(K,IJ)+UK(K,IJP))
      IF(CS(2).GE.0.)VLFR=0.5*(RLK(K,IJ)+RLK(K,IJT))*VK(K,IJ)*CS(2)
     &   *RB(I)
      IF(CS(2).LT.0.)VLFR=0.5*(RLK(K,IJR)+RLK(K,IJTR))*VK(K,IPJ)*CS(2)
     &    *RB(I)
      IF(FL(IMJ).NE.1) GO TO 1
      IF(FL(IJM).NE.1) GO TO 2
      GOTO3
    1 CS(3)=0.5*(UK(K,IMJ)+UK(K,IMJP))
      IF(CS(3).GE.0.)VLFL(K)=0.5*(RLK(K,IJL)+RLK(K,IJTL))*VK(K,IMJ)
     &   *CS(3)*RB(I-1)
      IF(CS(3).LT.0.)VLFL(K)=0.5*(RLK(K,IJ)+RLK(K,IJT))*VK(K,IJ)
     &   *CS(3)*RB(I-1)
      IF(FL(IJM).NE.1)GOTO2
      GOTO3
    2 CS(4)=0.5*(VK(K,IJM)+VK(K,IJ))
      IF(CS(4).GE.0.) VLFB(K,I)= 0.5*(RLK(K,IJ)+RLK(K,IJB))*VK(K,IJM)
     &   *CS(4)
      IF(CS(4).LT.0.) VLFB(K,I)= 0.5*(RLK(K,IJ)+RLK(K,IJT))*VK(K,IJ )
```

```
     &    *CS(4)
3     CONTINUE
      RETURN
      END
```

APPENDIX B

SOME PRELIMINARY RESULTS

Injection of Fines Into a Bed of Coarse Particles

The geometry of the system and the initial conditions are shown in Figure 9.1. At time t=0 a central at a velocity of V_g = 5 m/s turned on through an opening 1.27 cm wide. The jet contains fine particles (d_p = 80 μm, and ρ_p = 2.42 g/cm^3) at a volume fraction of 0.01. The fines enter at a velcoity of 16.67 cm/$_s$. Figure 9.2 shows the distribution fines in the bed of coarse particles. It is interesting to note the motion of packets of fines at a large concentration through the bed. It is being speculated that such simulations will be useful in situations such as injection of coal particles into a bed of inerts in a coal combustor.

Segregation Due to Size Difference

Figure 9.3 shows the initial conditions in a bed containing a uniform mixture of particles of two different sizes. At time t = 0 a central jet at a velocity of V_g = 1.27 m/s is turned on through an opening 1.27 cm wide. The volume fractions of both particulate phases at four locations in the bed are shown in Figure 9.4. These figures indicate that the mixture remains well mixed in the central jet region (locations 1 and 2). Far away from the jet (location 4) the particles tend to segregate.

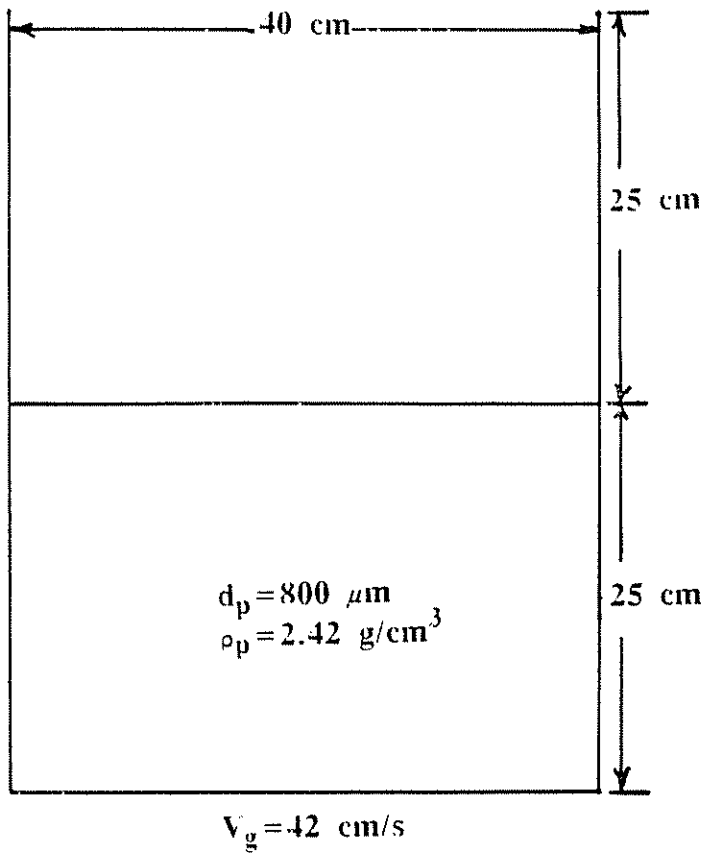
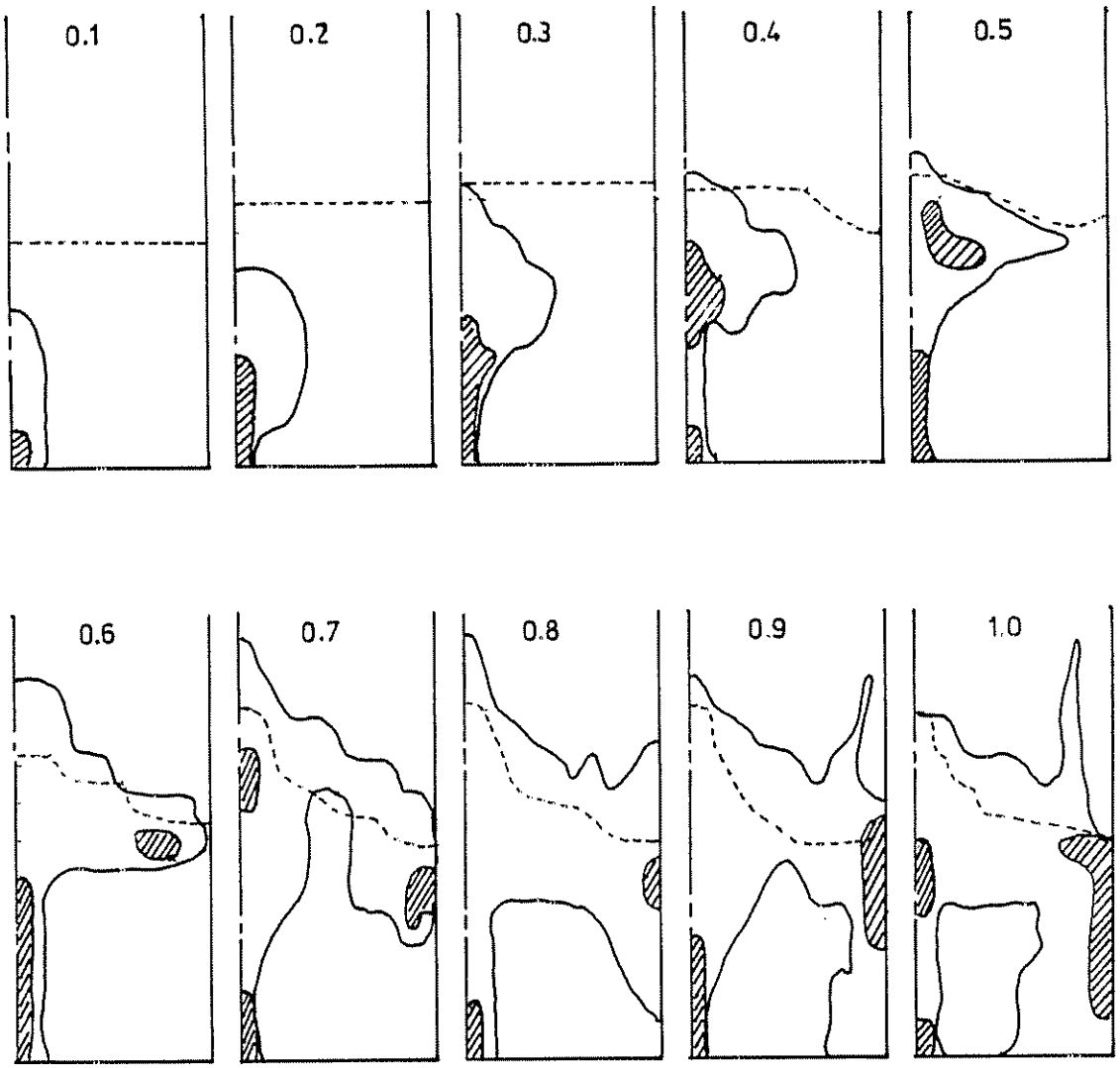


Figure 9.1. Initial Conditions for the Fines Injection Simulation



—— $\epsilon_2 > 10^{-4}$, ------ $\epsilon_1 > 0.2$, ▨ $\epsilon_2 > 10^{-3}$

$d_1 = 800\,\mu m$, $d_2 = 80\,\mu m$, $\rho_1 = \rho_2 = 2.42\ g/cm^3$

$V_{jet} = 5\ m/s$ $V_{grid} = 0.42\ m/s$ $W_2 = 0.77\ g/s\ cm$

Figure 9.2. Distributions of Fines Injected Through a Central Jet into a Bed of Coarse Particles

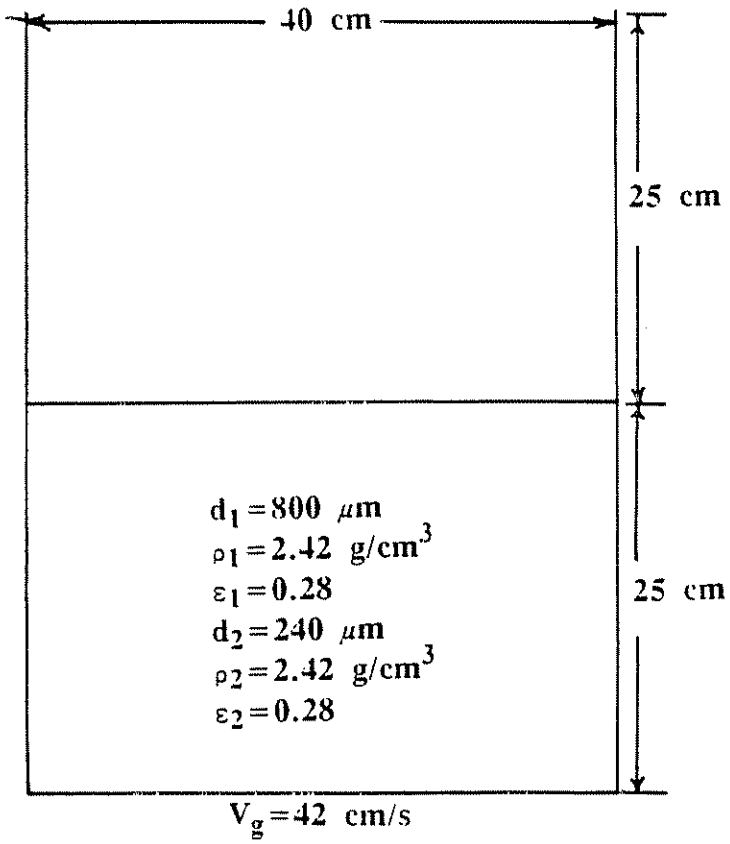


Figure 9.3. Initial Conditions for the Fines Injection Simulation

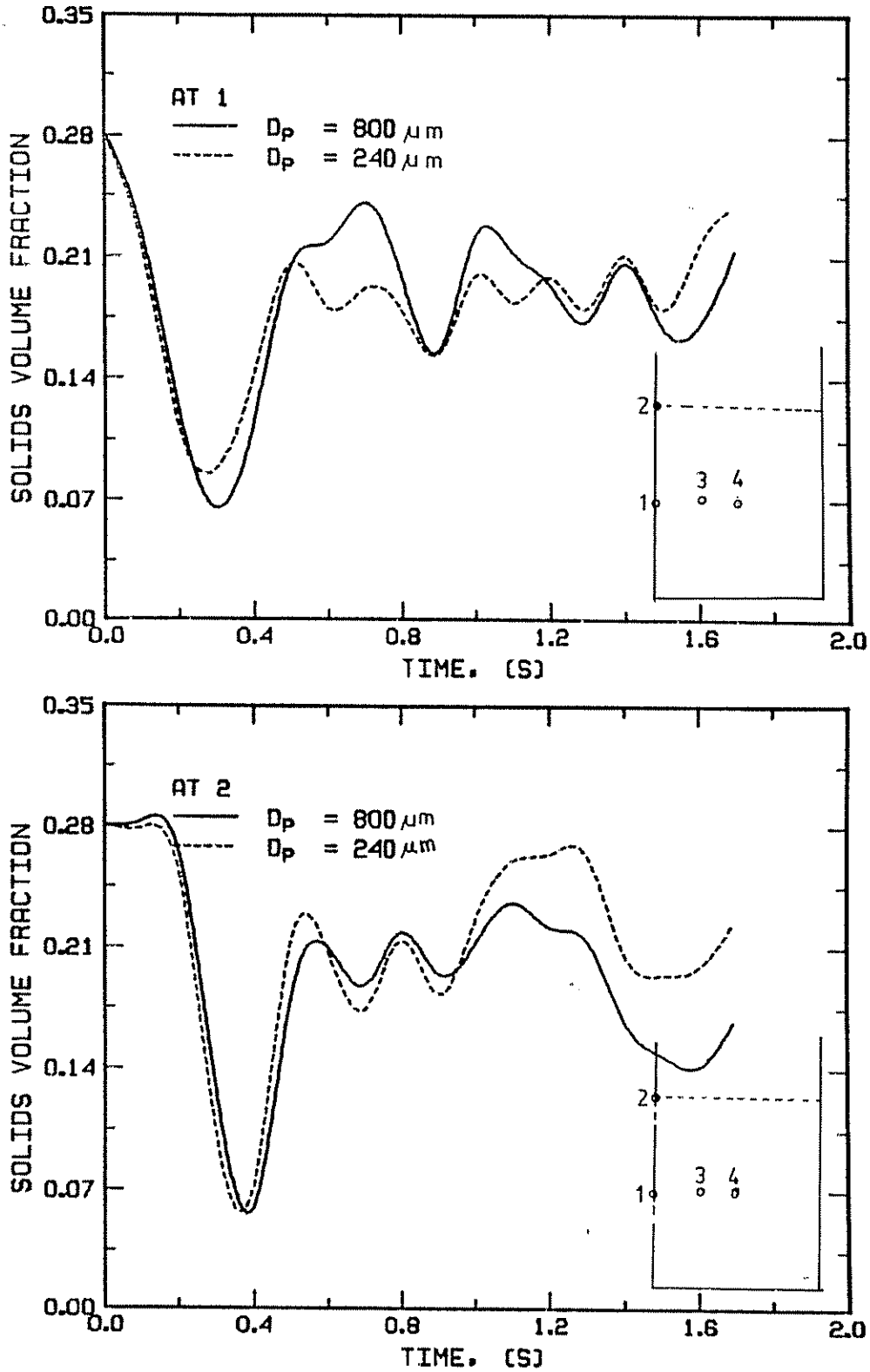


Figure 9.4. Volume Fractions of the Two Particulate Phases at Various Locations in the Bed

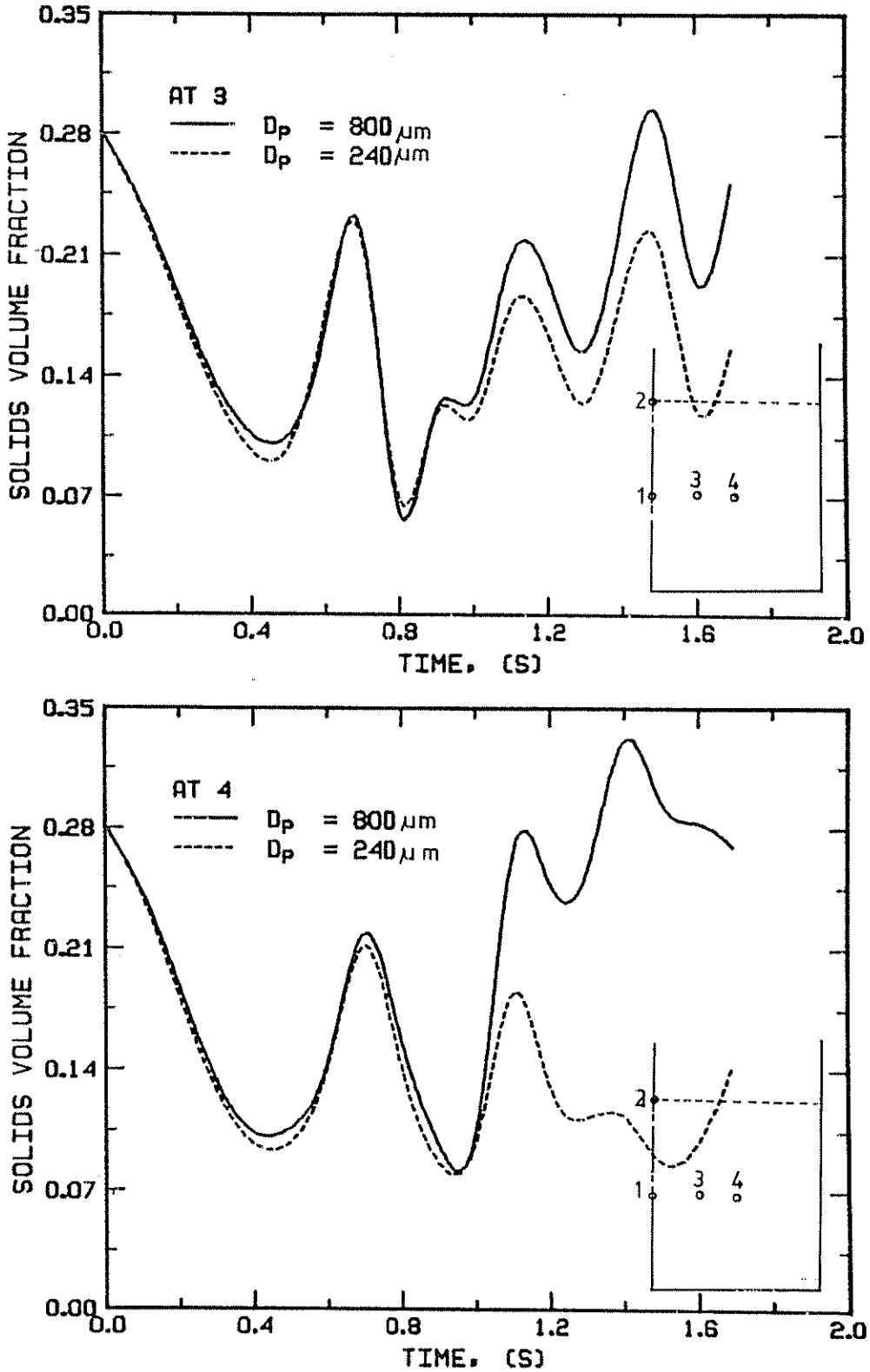


Figure 9.4. Volume Fractions of the Two Particulate Phases at Various Locations in the Bed

Mixing of Particles in a Fluidized Bed

The initial conditions for this run are shown in Figure 9.5. In an experiment starting with a uniform mixture of the particles indicated in the figure, a segregated layer of larger particles away from the central jet could be observed at a steady state. However, in the simulation the other extreme condition of two segregated layers was used as an initial condition. The volume fraction contours predicted using the simulation are shown in Figure 9.6. It can be seen from this figure that the bottom layer remained intact as a packed bed. Also, the bottom layer distributed the jet more evenly resulting in negligible mixing.

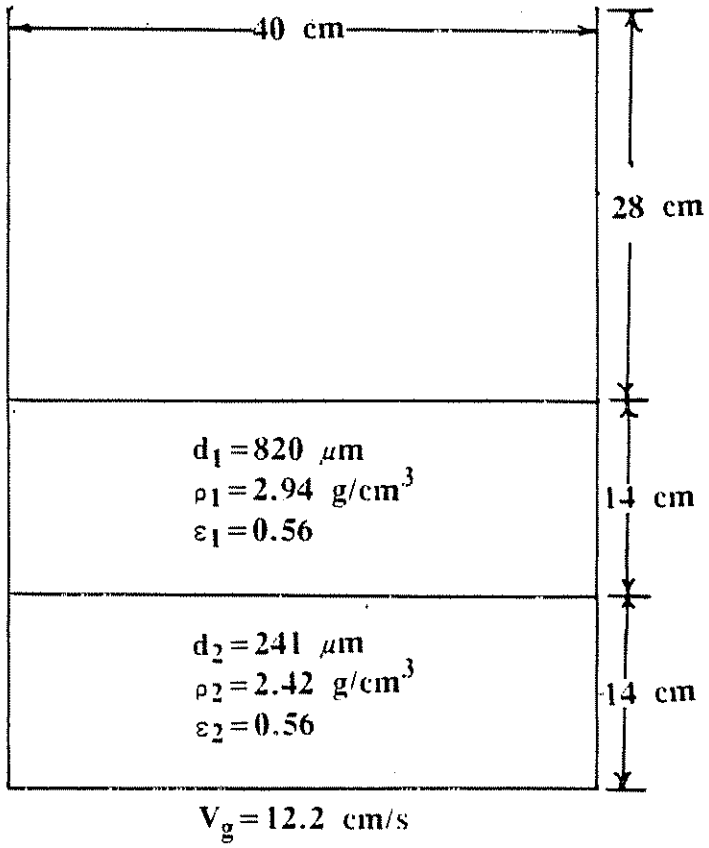


Figure 9.5. Initial Conditions for the Mixing Simulation

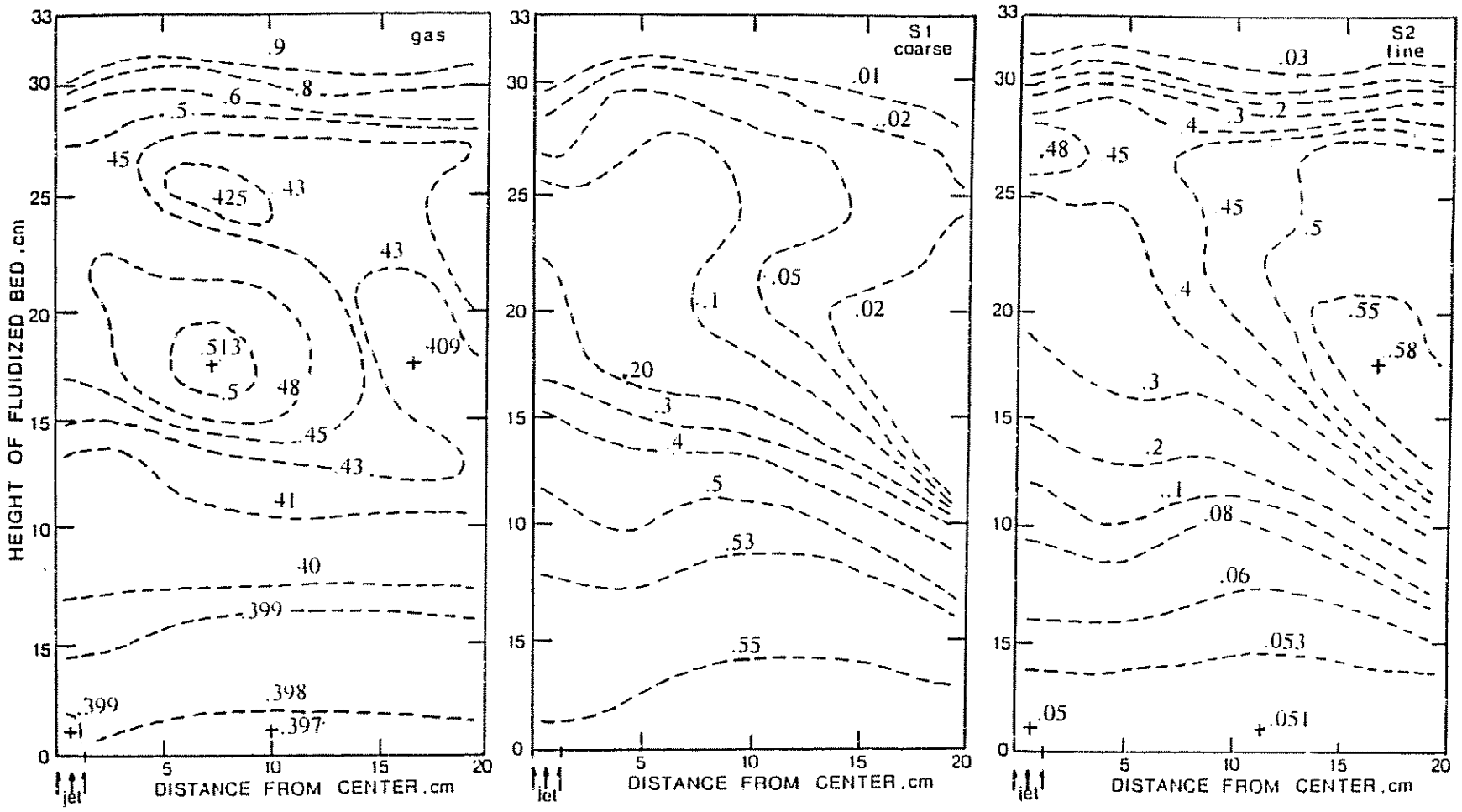


Figure 9.6. Solids Volume Fraction Contours at the End of Two Seconds in a Segregated Bed

BIBLIOGRAPHY

Adams, R. L., and Welty, J. R., "A Gas Convective Model of Heat Transfer in Large Particle Fluidized Beds," AIChE J. 25, 395-405 (1979).

Altiner, H. K. and J. F. Davidson, "Powder Flow From an Aerated Hopper," in "Fluidization" ed., Grace, J. R. and Matsen, J. M., Plenum Press, N.Y. 209-239 (1980).

Arastoopour, H. and Gidaspow, D., "Vertical Pneumatic Conveying Using Four Hydrodynamic Models," I&EC Fundamentals, 18, 123-130 (1979).

Arastoopour, H. and Gidaspow, D., "Analysis of IGT Pneumatic Conveying Data and Fast Fluidization Using a Thermohydrodynamic Model", Powder Technology, 22, 77-87 (1979).

Arastoopour, H., Lin, D. and Gidaspow, D., "Hydrodynamic Analysis of Pneumatic Transport of a Mixture of Two Particle Sizes," in "Multiphase Transport," Hemisphere Pub., 4, 1853 (1980).

Arastoopour, H., Lin, S.-C., and Weil, S.A., "Analysis of Vertical Pneumatic Conveying of Solids Using Multiphase Flow Models," AIChE J., 28, 467-473 (1982).

Arastoopour, H., Wang, C.-H. and Weil, S. A., "Particle-Particle Interaction Force in a Dilute Gas-Solid System," Chem. Eng. Science, 37, 1379-1384 (1982).

Bagnold, R. A., "Experiments on a Gravity-Free Dispersion of Large Solid Spheres in a Newtonian Fluid Under Shear," Proc. Roy. Soc. A225, 49-63 (1954).

Bauer, R. and Schlunder, E. U., "Effective Radial Thermal Conductivity of Packings in Gas Flow. Part II. Thermal Conductivity of the Packing Fraction Without Gas Flow," Int. Chem. Eng., 18, 189-204 (1978).

Beeckmans, J. M., Nilsson, J. and Large, J.-F., "Observations on the Mechanisms of Segregation in Flotsam-Rich, Fully Fluidized Beds," Ind. Eng. Chem. Fundam., 24, 90-95 (1985).

Boure, J. A., "Critical Two-Phase Flows" Chapter 10, in "Two Phase Flows and Heat Transfer with Application to Nuclear Reactor Design Problems" ed. J. J. Ginoux, von Karman Institute for Fluid Dynamics, Hemisphere Publishing Corp. (1978).

Botterill, J.S.M., "Fluid-Bed Heat Transfer," Academic Press, N.Y. (1975).

Chandok, S. S. and Pei, D.C.T., Ref. 71, S. L. Soo, "Multiphase Fluid Dynamics," Soo and Associates (1983).

Chiba, S., Chiba, T., Nienow, A.W., and Kobayoshi, H., "The Minimum Fluidization Velocity, Bed Expansion and Pressure-Drop Profile of Binary Particle Mixtures," Powder Tech., 22, 255-269 (1979).

Chakravarthy, S. R., "Euler Equations — Implicit Schemes and Boundary Conditions," AIAA J., 21, 5, 699-706 (1983).

Ettehadieh, B., "Hydrodynamic Analysis of Gas-Solid Fluidized Beds," Ph.D. Dissertation, Illinois Institute of Technology (1982).

Fedors, R. F. and Landel, R. F., "An Empirical Method of Estimating the Void Fraction in Mixtures of Uniform Particles of Different Size," Powder Technology, 23, 225-231 (1979).

Galloway, T. R. and Sage, B. H., "A Model of the Mechanism of Transport in Packed, Distended, and Fluidized Beds," Chem. Eng. Science, 25, 495-516 (1970).

Gelperin, N. I. and Einstein, V. G., "Heat Transfer in Fluidized Beds," in "Fluidization," Academic Press, N.Y. 471-540 (1971).

Gibillaro, L. G. and Rowe, P. N., "A Model for a Segregating Gas Fluidized Bed," Chem. Eng. Science, 29, 1403-1412 (1974).

Gidaspow, D., "Hyperbolic Compressible Two-Phase Flow Equations Based on Stationary Principles and the Fick's Law," in "Two-Phase Transport and Reactor Safety," ed. by Veziroglu, T. N. and Kakac, S., 1, 283-298, Hemisphere Publishing Corp. (1978).

Gidaspow, D., Lin, C. and Seo, Y. C., "Fluidization in Two Dimensional Beds With a Jet, Part I. Experimental Porosity Distributions," Ind. Eng. Chem. Fundam., 22, 184-193 (1983).

Gidaspow, D. and Ettehadieh, B., "Fluidization in Two-Dimensional Beds With a Jet 2. Hydrodynamic Modeling," I&EC Fundamentals, 22, 193-201 (1983).

Gidaspow, D., Syamlal, M., and Seo, Y.C., "Hydrodynamics of Fluidization: Supercomputer Modeling," presented at the AAAS meeting, Los Angeles, May 1985.

Gough, P.S., "Two-Dimensional Convective Flame-Spreading in Packed Beds of Granular Propellant," ARBL-CR-00404, Report to Ballistic Research Laboratory, Aberdeen Proving Ground, Marland, July 1979.

Greenspan, H. P. and Ungarish, M., "On Hindered Settling of Particles of Different Sizes," Int. J. Multiphase Flow, 8, 6, 587-604 (1982).

W. T. Hancox, Ferch, R. L., Liu, W. S. and Nieman, R. E., "One-Dimensional Models for Transient Gas-Liquid Flows in Ducts," Int. J. Multiphase Flow, 6, 25-40 (1980).

Harlow, F. H. and Amsden, A.A., "Numerical Calculation of Multiphase Fluid Flow," J. Comp. Phyics, 17, 19-52 (1975).

Hirt, C. W. and Nichols, B. D., "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," J. of Computational Physics, 39, 201-225 (1981).

Ishii, M., Thermo-Fluid Dynamic Theory of Two-Phase Flow, Eyrolles, France (1975).

Jackson, R., "The Mechanics of Fluidized Beds: Part I: The Stability of the State of Uniform Fluidization", Trans. Inst. Chem. Eng., 41, 13-21 (1963).

Jackson, R., "The mechamics of Fluidized Beds: Part II: The Motion of Fully Developed Bubbles," Trans. Inst. Chem. Eng., 41, 22-28 (1963).

Keairns, D. L. and Newby, R. A., "Fluidized Bed Gasifier Modeling," P. 107 in Ghate, M. and Martin, J. W., editors, "Coal Gasification Modeling Workshop Proceedings," DOE/METC/82-24, January 18-22, 1982, Morgantown, West Virginia.

Klein, H. H. and Scharff, M. F., "A Time Dependent Reactor Model of Fluidized Bed Chemical Reactors," ASME paper 82-FE-6 (1982).

Koo, J. H., and K.K. Kuo, "Transient Combution in Granulan Beds: Part I: Theoretical Modeling and Numerical Solution of Transient Combustion Processes in Mobile Granular Propellant Bed, Final Report 1974-1977, BRL-CR-346 AD-A044998 (Aug. 1977).

Kunii, D. and Levenspiel, O., Fluidization Engineering, John Wiley and Sons, Inc., New York (1969).

Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation," Comm. Pure and Applied Math., Vol. VII, 159-193 (1954).

Leung, L. S. and Wiles, R. J., Ind. Eng. Chem. Process Design and Development, 15, 552-557 (1976).

Lyczkowski, R. W., Gidaspow, D. and Solbrig, S. W., "Multiphase Flow-Models for Nuclear, Fossil and Biomass Energy Conversion," in Vol. II of Advances in Transport Processes, A. S. Majumdar, Editor, Wiley-Eastern Publisher 198-235 (1982).

Moretti, G., "Importance of Boundary Conditions in the Numerical Treatment of Hyperbolic Equations," The Physics of Fluids, 2 13-20 (1969).

Murray, J. D., "On the Mathematics of Fluidization", J. Fluid Mech., 21, 465-493 (1965).

Nakamura, K. and Capes, C. E., "Vertical Pneumatic Conveying of Binary Particle Mixtures," in "Fluidization Technology," 159-184, D. L. Keairns, ed., Hemisphere Publishing Corp., Washington (1976).

Nayak, A. L. and Tien, C. L., "A Statistical Thermodynamic Theory for Coordination-Number Distribution and Effective Thermal Conductivity of Random Packed Beds," Int. J. Heat, Mass Transfer 21, 669-676 (1978).

Ozkaynak, T. and Chen, J. C., "Emulsion Phase Residence Time and Its Use in Heat Transfer Models in Fluidized Beds," AIChE J. 26, 544-550 (1980).

Pai, S.-I., Two-Phase Flows, Vieweg Tracts in Pure and Applied Physics, Vol. 3, Braunschweig (1977).

Parrott, J. E. and Stuckes, A. D., "Thermal Conductivity of Solids," Methuen, Inc., N.Y. (1975).

Pigford, R. L. and Baron, T., Ind. Eng. Chem. Fundam., 4, 81-87 (1965).

Pillai, K. K., "Voidage Variation at the Wall of a Packed Bed of Spheres," Chem. Eng. Science, 32, 59-62 (1977).

Rasouli, F., Gidaspow, D. and Shin, Y. W., "Unequal Velocity Two Component Two Phase Flow by the Method of Characteristics and Critical Flow," Argonne National Laboratory Report (1981).

Rivard, W. C. and Torrey, M. D., "K-FIX: A Computer Program for Transient, Two-Dimensional, Two-Fluid Flow", LA-NUREG-6623 (Apr 1977).

Rowe, P. N., Partridge, B. A., Cheny, A. G., Henwood, G. A. and Lyall, E., "The Mechanisms of Solids Mixing in Fluidized Beds," Trans. Instn. Chem. Engrs., 43, 271-282 (1965).

Rowe, P. N. Niewnow, A. W. and Agbim, A. J., "A Preliminary Quantitative Study of Particle Segregation in Gas Fluidized Beds-Binary Systems of Near Spherical Particles," Trans. Instn. Chem. Engrs., 50, 324-333 (1972).

Rowe, P. N. and Nienow, A. W., "Minimum Fluidisation Velocity of Multi-Component Particle Mixtures," Chem. Eng. Science, 30, 1365-1369 (1975).

Rowe, P. N. and Niewnow, A. W., "Particle Mixing and Segregation in Gas Fluidised Beds. A Review," Powder Technology, 15, 141-147 (1976).

Ruckenstein, E. and Tzeculescu, M., "Proceedings of the International Symposium on Fluidzation in Eindhoven," Drinkenburg, A.A.H., ed., Netherlands University Press, 180, 1967/1968.

Saxena, S. C., Grewal, N. S., Gabor, J. D., Zabrodsky, S. S. and Galershtein, D. M., "Heat Transfer Between a Gas Fluidized Bed and Immersed Tubes," in "Advances in Heat Transfer," 14, Academic Press, N.Y. 149-159 (1978).

Scharff, M., Chan, R.K.-C., Chiou, M. J., Dietrich, D. E., Dion, D. D., Klein, H. H., Laird, D. N., Levine, H. R., Meister, C. A., Srinivas, B., "Computer Modeling of Mixing and Agglomeration in Coal Conversion Reactors," VOl. I & II, DOE/ET/10329-1211, JAYCOR, (Feb. 1982).

Schneyer, G. P., Peterson, E. W., Chen, P. J., Cook, J. L., Brownell, D. H. and Blake, T. R., "Computer Modeling of Coal Gasification Reactors," Final Report for June 1975-1980 DOE/ET/10247 Systems, Science, Software (April 1981).

Shi, Y.-F. and Fan, L. T., "Lateral Mixing of Solids in Batch Gas-Solids Fluidized Beds," Ind. Eng. Chem. Process Des. Dev., 23, 337-341 (1984).

Soo, S. L., "Fluid Dynamics of Multiphase Systems," Blaisdell Publishing Corp., Waltham, MA, 1967.

Trapp, J. A. Ransom, V. H., "A Choked-Flow Calculation Criterion for Nonhomogeneous, Nonequilibrium, Two-Phase Flows," Int. J. Multiphase Flow, 8, 669-681 (1982).

Valenzuela, J. A., "Mixing of Solids in Bubbling Fluidized Beds and its Application to Fluidized Bed Combustors," ScD Dissertation, Massachusetts Institute of Technology, 1982.

Vollheim, R., "Behavior of the Wall Shear Stress of the Conveying Medium in a Pneumatic Conveyor and its Consequences to the Heat Transfer", (Translation), Maschinenbautechnik, 12, 233-236 (1963).

Wakao, N., Kaguei, S. and Funazkri, T., "Effect of Fluid Dispersion Coefficients on Particle-to-Fluid Heat Transfer Coefficients in Packed Beds," Chem. Eng. Science 34, 325-332 (1979).

Wen, C. Y. and Galli, A.S. "Dilute Phase System in Fluidization", ed. Davidson, J. F. and Harrison, D., Academic Press, London, Chapter 16, p. 677-710 (1971).

Wen, C. Y. and Yu, Y. H., "Mechanics of Fluidzation," Chem. Eng. Prog. Symp. Series No. 62, 62, 100 (1966).

Yang, W.-C., and Keairns, D. L., "Rate of Particle Separation in a Gas Fluidized Bed," Ind. Eng. Chem. Fundam., 21, 228-235 (1982).

Zabrodsky, S. S., "Hydrodynamics and Heat Transfer in Fluidized Beds," The M.I.T. Press, Cambridge, MA (1966).

Zabrodsky, S. S., Epanov, Yu. G. and Galershtein, D. M., "On the Dependence of Fluidized Bed-Wall Heat Transfer Coefficients on the Thermal Conductivity and Volumetric Heat Capacity of the Particles," in "Fluidization," Cambridge University Press 362-370 (1978).